

Patent Abstracts of Japan

PUBLICATION NUMBER : 11345193

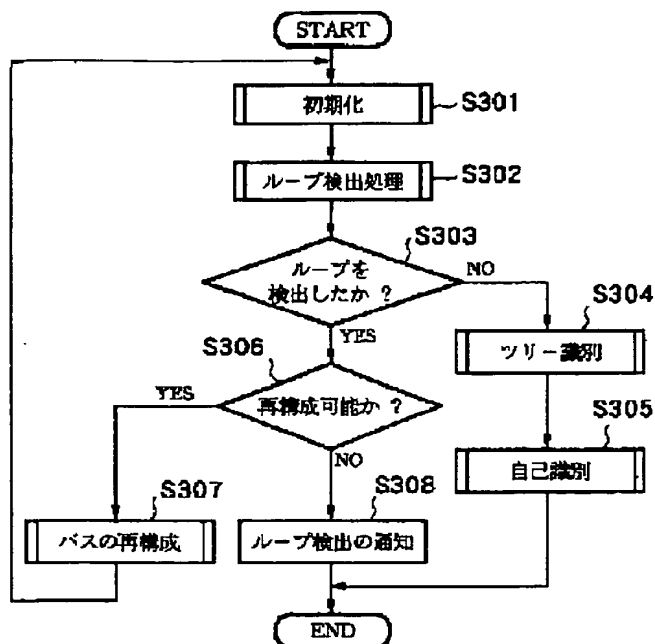
PUBLICATION DATE : 14-12-99

APPLICATION DATE : 29-05-98

APPLICATION NUMBER : 10150222

APPLICANT : CANON INC;

INVENTOR : TSUJIMOTO TAKUYA;

INT.CL. : G06F 13/14 B41J 29/38 G06F 13/00
G06F 13/00 G06F 13/38 H04L 12/28TITLE : SERIAL BUS INTERFACE DEVICE,
CONSTITUTING METHOD FOR BUS,
RECORDING MEDIUM, AND SERIAL
BUS INTERFACE SYSTEM

ABSTRACT : PROBLEM TO BE SOLVED: To solve the problem that the exchange of a signal for identifying a tree structure is disabled when non-permitted connection configuration is adopted in connection between devices, resulting in data communication between the devices becoming unavailable.

SOLUTION: The additional connection or disconnection of a device is detected, a bus is constituted and when constituting the bus, the existence of a loop-connected device on an interface system is detected. A parent notify signal is forcedly generated from a port to be the port of connection with another device, and the parent notify signal is received from the port. The spot of loop connection generation is decided from the transmission and reception conditions of the signal and based on the decision result, active ports are disabled. After the port is disabled, the bus constitution is restarted.

COPYRIGHT: (C)1999,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-345193

(43) 公開日 平成11年(1999)12月14日

(51) Int.Cl. ⁸	識別記号	F I	
G 0 6 F 13/14	3 3 0	G 0 6 F 13/14	3 3 0 E
B 4 1 J 29/38		B 4 1 J 29/38	Z
G 0 6 F 13/00	3 5 1	G 0 6 F 13/00	3 5 1 M
	3 5 7		3 5 7 A
13/38	3 5 0	13/38	3 5 0
審査請求 未請求 請求項の数30 O L (全 35 頁) 最終頁に続く			

(21) 出願番号 特願平10-150222

(22) 出願日 平成10年(1998) 5 月29日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 辻本 卓哉

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

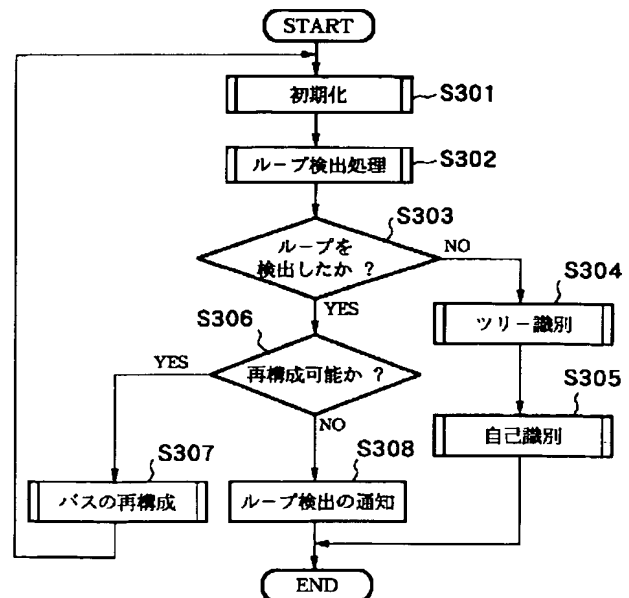
(74) 代理人 弁理士 大塚 康德 (外2名)

(54) 【発明の名称】 シリアルバスインタフェースデバイスおよびバスの構成方法、記録媒体、シリアルバスインタフェースシステム

(57) 【要約】

【課題】 デバイス間接続で、許可されない接続構成がとられると、ツリー構造を識別するための信号の授受ができず、デバイス間のデータ通信ができなくなる。

【解決手段】 デバイスの追加接続や切離しを検知し、バスの構成を行い、バス構成時にインタフェースシステム上におけるループ接続されたデバイスの存在を検知する。他のデバイスとの接続口となるポートから強制的に Parent Notify信号を発生させ、前記ポートから Parent Notify信号を受信する。信号の送信および受信状況からループ接続の発生箇所を判断し、その判断結果に基づきアクティブなポートをディセーブルにする。ポートのディセーブル後に前記バス構成を再び開始することで上記課題を解決した。



【特許請求の範囲】

【請求項1】 デバイスの追加接続や切離しを検知しバスの構成を行う構成手段と、
インタフェースシステム上でデバイスの接続形態を判断し、ループ接続されたデバイスの存在を検知するループ検知手段と、
他のデバイスとの接続口となるひとつもしくはそれ以上の数のポートと、
前記ポートから強制的にParent Notify信号を発生させる信号発生手段と、
前記ポートからParent Notify信号を受信する信号受信手段と、
前記信号の送信および受信状況からループ接続の発生箇所を判断する判断手段と、
前記判断手段の判断結果に基づきアクティブなポートをディセーブルにするディセーブル手段と、
前記ディセーブル手段によるポートのディセーブル後に前記バス構成手段を再び開始させるシステム再構成開始手段と、
を備えることを特徴とするシリアルバスインタフェースデバイス。

【請求項2】 前記シリアルバスインタフェースが、IEEE1394に準拠したものであることを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項3】 前記ループ検知手段は、タイマーによるカウント値が任意のタイマー値を超えた場合、バス上にループ接続されたデバイスが存在すると認識することを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項4】 前記ループ検知手段によるループの検知結果を、ユーザーに通知する手段を更に備えることを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項5】 前記ループ検知手段によるループの検知結果をユーザーに通知する手段は、表示装置の表示に基づくものであることを特徴とする請求項4に記載のシリアルバスインタフェースデバイス。

【請求項6】 前記シリアルバスインタフェースデバイスがプリンタで、メディアに対する印字を可能とするデバイスである場合、前記ループ検知手段によるループの検知結果をメディアに印字出力することでユーザーに通知することを特徴とする請求項4に記載のシリアルバスインタフェースデバイス。

【請求項7】 前記ループ検知手段によるループの検知結果をユーザーに通知する手段は、音声伝達に基づくものであることを特徴とする請求項4に記載のシリアルバスインタフェースデバイス。

【請求項8】 前記ポートが複数である場合、複数のポートそれぞれでループ検知が可能であることを特徴とする請求項1に記載のシリアルバスインタフェースデバイ

ス。

【請求項9】 前記判断手段によって、前記シリアルバスインタフェースデバイスがシリアルバスインタフェースシステム上のループを構成しているバス上に存在すると判断された場合、前記システム再構成開始手段はポートをディセーブル後、システム再構成を実行し、バス上に存在しないと判断された場合、前記表示手段はループ接続の存在のみをユーザーに通知することを特徴とする請求項1または4に記載のシリアルバスインタフェースデバイス。

【請求項10】 前記判断手段は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させ、送信したのとは別のポートでParent Notify信号を受信したときに、そのポート間に接続されるバス上でループが存在すると判断することを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項11】 前記信号発生手段は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させても、他のどのポートからもParent Notify信号を受信できないときには、さらにParent Notify信号を受信していない別のポートに対してParent Notify信号を発生し、ループの存在を確認する動作を行うことを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項12】 前記判断手段は、アクティブなポートが存在する場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させたときに、任意のタイマー値でタイマーを起動し、タイマー値を超えた時点で他のどのポートからもParent Notify信号を受信できないときそのポートの先に接続されるデバイス間で構成されるバス上でループが存在すると判断することを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項13】 前記ループ検知手段で用いられる任意のタイマー値と、前記判断手段で用いられる任意のタイマー値とは、別個の値が設定され、前記ループ検知手段と前記判断手段の起動タイミングを変えていることを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項14】 前記ディセーブル手段によってポートがディセーブルにされた場合、どのポートがディセーブルにされたかユーザーに通知する通知手段を更に備えることを特徴とする請求項1に記載のシリアルバスインタフェースデバイス。

【請求項15】 デバイスの追加接続や切離しを検知しバスの構成を行うバス構成工程と、
バス構成時インタフェースシステム上におけるループ接続されたデバイスの存在を検知するループ検知工程と、
他のデバイスとの接続口となるポートから強制的にParent

nt Notify信号を発生させる信号発生工程と、
前記ポートからParent Notify信号を受信する信号受信工程と、
前記信号の送信および受信状況からループ接続の発生箇所を判断する判断工程と、
前記判断工程の判断結果に基づきアクティブなポートをディセーブルにするディセーブル工程と、
前記ディセーブル工程によるポートのディセーブル後に前記バス構成工程を再び開始させるシステム再構成開始工程と、
を有することを特徴とするシリアルバスインタフェースのバス構成方法。

【請求項16】 前記ループ検知工程は、タイマーによるカウント値が任意のタイマー値を超えた場合、バス上にループ接続されたデバイスが存在すると認識することを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項17】 前記ループ検知工程によるループの検知結果を、ユーザーに通知する工程を更に備えることを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項18】 前記ユーザーに通知する工程は、ディスプレイにより、前記ループ検知工程によるループの検知結果を通知することを特徴とする請求項17に記載のシリアルバスインタフェースのバス構成方法。

【請求項19】 前記シリアルバスインタフェースバスを構成するデバイスがプリンタ等のメディアに対する印字を可能とするデバイスの場合、前記ループ検知工程によるループの検知結果をメディアに印字出力する方法でユーザーに通知することを特徴とする請求項17に記載のシリアルバスインタフェースのバス構成方法。

【請求項20】 前記ユーザーに通知する工程は、音声伝達により、前記ループ検知工程によるループの検知結果を通知することを特徴とする請求項17に記載のシリアルバスインタフェースのバス構成方法。

【請求項21】 前記ループ検知工程は、複数のループを検知可能なことを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項22】 前記判断工程によって、前記シリアルバスインタフェースデバイスがシリアルバスインタフェースシステム上のループを構成しているバス上に存在すると判断された場合、前記システム再構成開始工程はポートをディセーブル後、システム再構成を実行し、バス上に存在しないと判断された場合、前記ユーザーに通知する工程はループ接続の存在のみをユーザーに通知することを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項23】 前記判断工程は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させ、送信した

のとは別のポートでParent Notify信号を受信したときに、そのポート間に接続されるバス上でループが存在すると判断することを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項24】 前記信号発生工程は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させても、他のどのポートからもParent Notify信号を受信できないときには、さらにParent Notify信号を受信していない別のポートに対してParent Notify信号を発生し、ループの存在を確認する動作を行うことを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項25】 前記判断工程は、アクティブなポートが存在する場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させたときに、任意のタイマー値でタイマーを起動し、タイマー値を超えた時点で他のどのポートからもParent Notify信号を受信できないときそのポートの先に接続されるデバイス間で構成されるバス上でループが存在すると判断することを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項26】 前記ループ検知工程で用いられる任意のタイマー値と、前記判断工程で用いられる任意のタイマー値とは、別個の値が設定され、前記ループ検知工程と前記判断工程の起動タイミングを変えていることを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項27】 前記ディセーブル工程によってポートがディセーブルにされた場合、どのポートがディセーブルにされたかユーザーに通知する通知工程を更に備えることを特徴とする請求項15に記載のシリアルバスインタフェースのバス構成方法。

【請求項28】 デバイスの追加接続や切離しを検知しバスの構成を行うバス構成工程と、
バス構成時インタフェースシステム上におけるループ接続されたデバイスの存在を検知するループ検知工程と、
他のデバイスとの接続口となるポートから強制的にParent Notify信号を発生させる信号発生工程と、

前記ポートからParent Notify信号を受信する信号受信工程と、
前記信号の送信および受信状況からループ接続の発生箇所を判断する判断工程と、
前記判断工程の判断結果に基づきアクティブなポートをディセーブルにするディセーブル工程と、
前記ディセーブル工程によるポートのディセーブル後に前記バス構成工程を再び開始させるシステム再構成開始工程と、
をコンピュータに実行させるためのプログラムを記録したことを特徴とする記録媒体。

【請求項29】 複数のデバイスがシリアルバスインタフェースデバイスで接続されて構成されるシリアルバスインタフェースシステムにおいて、前記シリアルバスインタフェースデバイスとして、請求項1記載のシリアルバスインタフェースデバイスを用いたことを特徴とするシリアルバスインタフェースシステム。

【請求項30】 前記シリアルバスインタフェースデバイスで接続されたデバイスはプリンタ機能を備えることを特徴とする請求項29記載のシリアルバスインタフェースシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、シリアルバスインタフェースに関し、特にデバイス及びバスの構成方法、記録媒体、該デバイスを使用したシリアルバスインタフェースシステムに関するものである。

【0002】

【従来の技術】近年シリアルバスが、信号線が少ないこと、ケーブルが細いこと、コネクタが小さいこと、IDやターミネータ等の設定が不要なこと、活線挿抜が可能

20

なこと、等時性のあるデータ転送（アイソクロナス転送）が可能なことなどの優れた特徴をもつことから脚光を浴びてきている。

【0003】特にIEEE1394のシリアルバスは、動画像等の大容量データの高速伝送が可能であること、バスアーキテクチャによってメモリアクセスが可能であること、ホットプラグインやプラグアンドプレイが可能であること、およびピア・ツー・ピア接続が可能であることなどの特徴を持つことによって、パーソナルコンピュータだけでなく家庭内のAV機器（デジタルビデオカメラ等）やそれ以外の機器への適用が盛んに進められている。

【0004】また、更なる高速化や長距離化に対応した新たな規格の策定も現在精力的に進められている。

【0005】ここでIEEE1394バスの構成例を図17を用いて説明する。

【0006】1701～1708は、IEEE1394インターフェースでディジーチェーンあるいはブランチで接続されているデバイス（ノード）である。デバイスとしては、コンピュータ、デジタルビデオカメラ、CCDカメラ、プリンタ、スキャナ、ストレージデバイス等様々なデバイスが考えられる。IEEE1394のポートは、2組のツイストペアケーブル（一方をA、他方をBと称する）と1組の電源ペアケーブルの計6本のケーブルで構成され、2組のツイストペアケーブルはお互いケーブルでクロスして接続され、一方のAは他方のBに、一方のBは他方のAに接続される。データ信号は半二重通信で、ツイストペアAでデータ信号を、ツイストペアBでストローブ信号を差動信号として送信し、ツイストペアAでストローブ信号を、ツイストペアBでデータ信号を差動信

40

50

号として受信する。また、調停信号は全二重通信で、ドライバ側から調停信号を送信し、同時にレシーバ側ではある電圧レベルによって定義された3相（0, 1, Z）の論理を用いて受信した調停信号を識別する。上記の3相の論理を判別するために、ツイストペアAには「Ar b Rx」として、ツイストペアBには「Ar b B Rx」として非反転入力と反転入力を1組にしたコンパレータが用意されている。「0」は非反転入力での低電圧と反転入力での高電圧電圧と定義され、「1」は非反転入力での高電圧と反転入力での低電圧と定義され、「Z」は非反転と反転の両入力での低電圧で定義されている。

【0007】バスの構成を確立するには、バスの初期化、ツリー識別、自己識別の3つのフェーズが実施される。この3つのフェーズのプロセスの間に、図17に示したようなツリーに似たトポロジーが構築される。

【0008】ここでバスの初期化は、電源不連続で発生する場合や、ソフトウェアコマンドによって発生する場合や、またはバスの構成でノードの新規接続や切り離し等によるバス構成の不調を検知した場合によりバスリセットが発生し、すべてのノードがアイドル状態になることにより完了する。

【0009】具体的には、まずツイストペアAとツイストペアBの両方に論理「1」を送信してバスリセットを発生し、この信号を受信したノードはバスリセットの発生を認識する。バスリセット発生後任意の時間が経過すると、ツイストペアAとツイストペアBの両方に論理「Z」を送信してアイドル信号を発生させ、接続されている相手ノードがアイドル信号を送信するのを待ち、相手ノードからアイドル信号を受信するとバスの初期化が完了したことを認識する。

【0010】バスの初期化が完了すると、次にツリー識別のフェーズに移行する。

【0011】ノードには、自ノードに対する接続ノードが1つだけ、つまりバス上では端点として構成される

「Leaf」ノードと、2つ以上の接続ノードを持つ「Branch」ノードと、接続ノードを全く持たない、つまりスタンドアローンで構成される「非接続」ノードの3種類のノードがある。

【0012】ツリー識別では、まず最初にノードが親（ルート）候補であるノードに対してParent Notify信号を送信する。図17では、「Leaf」ノードである1704, 1706, 1707, 1708の各ノードからParent Notify信号が送信されていることを示している。

「Branch」ノードでは、複数のポートに他のデバイスが接続されている場合にどのポート（そのポートの先に接続されているノード）がルート候補のノードであるのか認識できないので、最初にParent Notify信号を送信することはなく、「Leaf」ノードからのParent Notify信号を受信するのを待つことになる。ただし、「Branch」

ノードで複数ポートを有していても他のノードと接続されているポートが1ポートしかなく他のポートがすべて非接続の場合、もしくは接続されているノードが複数であってもアクティブになっているポート（もしくはそのポートの接続先のノード）が一つしかない場合には「Leaf」ノードになる。

【0013】次に、図18で示されるように、Parent Notify信号を受信した「Branch」ノードが送信してきたノードに対して子ノードであることを認めるChild notify信号を送信する。ここで1801～1808はノードである。「Leaf」ノードはChild notify信号を受信しそのポートの接続が完了したことを認識する。具体的には、ノード1804からのParent Notify信号を受信したノード1802は、ノード1804に対してChild notify信号を送信し、その信号を受信したノード1804はノード1802との接続が正常に行われていることを認識する。同様にノード1807、ノード1808とノード1805の間で、それからノード1806とノード1803の間でも同じことが行われる。

【0014】「Branch」ノードは、上述したようにParent Notify信号に対してChild notify信号を送信し、そのポート間でのノードの接続が完了したことを認識した後に、自ノードのポートでアクティブとなっているポートのうちParent Notify信号を受信していないポートが残り1ポートとなると、そのポートに対してParent Notify信号を送信する。

【0015】図18においては、ノード1803および1804がその動作を行っている。具体的には、ノード1803は3つのポートを持つノードで、アクティブなポートを二つ持ち、そのうち一つのポートでの接続が完了（ノード1806との接続）したため残り一つのポート（接続ノードは1801）に対してParent Notify信号を送信する。同様にノード1805もノード1802に対してParent Notify信号を送信する。その結果ノード1801とノード1802が最終的な親（ルート）候補となる。

【0016】図19では、ノード1901がノード1902に対してParent Notify信号を送信し、ノード1902がノード1901に対してChild notify信号を送信することで、最終的な親である「Root」ノードにノード1902が確定しツリー識別が完了したことを示している。ここで1901～1908はノードである。

【0017】ツリー識別が完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別のフェーズに入る。

【0018】

【発明が解決しようとする課題】しかしながら上述した従来の技術においては、他のデバイスと接続を行うためのケーブルのコネクタがすべて同一であることから、図20に示されるようにノード2003とノード2005

が接続されると、ディジーチェーン接続やブランチ接続以外の許容されていないループ接続がノード2001、ノード2002、ノード2003およびノード2005との間で成立してしまうことになる。

【0019】このループ接続では、「Branch」ノードの複数のポートが、他ノードからのParent Notify信号を受信するのを待つため、このノードからParent Notify信号を送信することはなくツリー識別を完了させることができない。そのため、その後の自己識別やデータ通信ができない状態に陥ってしまうという問題がある。

【0020】この問題を解決する手段としてループ接続が発生した場合、それを検知して何らかの手段でユーザーに通知するという方法もあるが、検知手段がタイマーによる時間管理、つまりある一定時間内にツリー識別のフェーズが終了しなかった場合にループ接続がバス上で検知されたと見なすものであれば、一体どの箇所でループ接続が発生したのかかわからないし、ループ接続以外の要件によりタイムアウトする場合も考慮すれば確実にループが存在するかをこれだけの情報で判断することは難しい（つまりタイマーによる時間管理では確実なループ検知はできない）。

【0021】また、構成されるバスの中に複数のループが存在する場合もその検知はできないし（ループ接続が存在することはわかっていてもいくつのループがあるかまではわからない）、できたとしてもその箇所を特定することはできない。さらに、検知した場所を的確に伝えられたとしてもその情報から正常なバスの構成を行うにはユーザー自らがノードの切り離しなどによるループ接続の回避作業を行わなければならないなど様々な問題がある。

【0022】本発明は上記問題を解決するものであり、確実なループ検知を行いそのループ接続を自動的に回避してバスを再構成する。また再構成できない場合も含めてループ接続の発生箇所をユーザーに対して的確に知らせることができる。

【0023】

【課題を解決するための手段】上記目的を達成するために本発明にかかるシリアルバスインタフェースデバイス及びシリアルバスインタフェースのバス構成方法、記録媒体、該デバイスを使用したシステムは以下のような構成になる。

【0024】すなわち、シリアルバスインタフェースデバイスは、デバイスの追加接続や切離しを検知しバスの構成を行う構成手段と、インタフェースシステム上でデバイスの接続形態を判断し、ループ接続されたデバイスの存在を検知するループ検知手段と、他のデバイスとの接続口となるひとつもしくはそれ以上の数のポートと、前記ポートから強制的にParent Notify信号を発生させる信号発生手段と、前記ポートからParent Notify信号を受信する信号受信手段と、前記信号の送信および受信

状況からループ接続の発生箇所を判断する判断手段と、前記判断手段の判断結果に基づきアクティブなポートをディセーブルにするディセーブル手段と、前記ディセーブル手段によるポートのディセーブル後に前記バス構成手段を再び開始させるシステム再構成開始手段とを備える。

【0025】また好ましくは、前記シリアルバスインタフェースは、IEEE1394に準拠したものである。

【0026】また好ましくは、前記ループ検知手段は、タイマーによるカウント値が任意のタイマー値を超えた場合、バス上にループ接続されたデバイスが存在すると認識する。

【0027】また好ましくは、前記ループ検知手段によるループの検知結果を、ユーザーに通知する手段を更に備える。

【0028】また好ましくは、前記ループ検知手段によるループの検知結果をユーザーに通知する手段は、表示装置の表示に基づくものである。

【0029】また好ましくは、前記シリアルバスインタフェースデバイスがプリンタで、メディアに対する印字を可能とするデバイスである場合、前記ループ検知手段によるループの検知結果をメディアに印字出力することでユーザーに通知する。

【0030】また好ましくは、前記ループ検知手段によるループの検知結果をユーザーに通知する手段は、音声伝達に基づくものである。

【0031】また好ましくは、前記ポートが複数である場合、複数のポートそれぞれでループ検知が可能である。

【0032】また好ましくは、前記判断手段によって、前記シリアルバスインタフェースデバイスがシリアルバスインタフェースシステム上のループを構成しているバス上に存在すると判断された場合、前記システム再構成開始手段はポートをディセーブル後、システム再構成を実行し、バス上に存在しないと判断された場合、前記表示手段はループ接続の存在のみをユーザーに通知する。

【0033】また好ましくは、前記判断手段は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させ、送信したのとは別のポートでParent Notify信号を受信したときに、そのポート間に接続されるバス上でループが存在すると判断する。

【0034】また好ましくは、前記信号発生手段は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させても、他のどのポートからもParent Notify信号を受信できないときには、さらにParent Notify信号を受信していない別のポートに対してParent Notify信号を発生し、ループの存在を確認する動作を行う。

【0035】また好ましくは、前記判断手段は、アクテ

ィブなポートが存在する場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させたときに、任意のタイマー値でタイマーを起動し、タイマー値を超えた時点で他のどのポートからもParent Notify信号を受信できないときそのポートの先に接続されるデバイス間で構成されるバス上でループが存在すると判断する。

【0036】また好ましくは、前記ループ検知手段で用いられる任意のタイマー値と、前記判断手段で用いられる任意のタイマー値とは、別個の値が設定され、前記ループ検知手段と前記判断手段の起動タイミングを変えていることを特徴とする。

【0037】また好ましくは、前記ディセーブル手段によってポートがディセーブルにされた場合、どのポートがディセーブルにされたかユーザーに通知する通知手段を更に備える。

【0038】また、インタフェースシステムのバス構成方法は、デバイスの追加接続や切離しを検知しバスの構成を行うバス構成工程と、バス構成時インタフェースシステム上におけるループ接続されたデバイスの存在を検知するループ検知工程と、他のデバイスとの接続口となるポートから強制的にParent Notify信号を発生させる信号発生工程と、前記ポートからParent Notify信号を受信する信号受信工程と、前記信号の送信および受信状況からループ接続の発生箇所を判断する判断工程と、前記判断工程の判断結果に基づきアクティブなポートをディセーブルにするディセーブル工程と、前記ディセーブル工程によるポートのディセーブル後に前記バス構成工程を再び開始させるシステム再構成開始工程とを有する。

【0039】また好ましくは、前記ループ検知工程は、タイマーによるカウント値が任意のタイマー値を超えた場合、バス上にループ接続されたデバイスが存在すると認識する。

【0040】また好ましくは、前記ループ検知工程によるループの検知結果を、ユーザーに通知する工程を更に備える。

【0041】また好ましくは、前記ユーザーに通知する工程は、ディスプレイにより、前記ループ検知工程によるループの検知結果を通知する。

【0042】また好ましくは、前記シリアルバスインタフェースバスを構成するデバイスがプリンタ等のメディアに対する印字を可能とするデバイスの場合、前記ループ検知工程によるループの検知結果をメディアに印字出力する方法でユーザーに通知する。

【0043】また好ましくは、前記ユーザーに通知する工程は、音声伝達により、前記ループ検知工程によるループの検知結果を通知する。

【0044】また好ましくは、前記ループ検知工程は、複数のループを検知可能である。

【0045】また好ましくは、前記判断工程によって、前記シリアルバスインタフェースデバイスがシリアルバスインタフェースシステム上のループを構成しているバス上に存在すると判断された場合、前記システム再構成開始工程はポートをディセーブル後、システム再構成を実行し、バス上に存在しないと判断された場合、前記ユーザに通知する工程はループ接続の存在のみをユーザに通知する。

【0046】また好ましくは、前記判断工程は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させ、送信したのとは別のポートでParent Notify信号を受信したときに、そのポート間に接続されるバス上でループが存在すると判断する。

【0047】また好ましくは、前記信号発生工程は、複数のポートがアクティブの場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させても、他のどのポートからもParent Notify信号を受信できないときには、さらにParent Notify信号を受信していない別のポートに対してParent Notify信号を発生し、ループの存在を確認する動作を行う。

【0048】また好ましくは、前記判断工程は、アクティブなポートが存在する場合にParent Notify信号を受信していないポートに対してParent Notify信号を発生させたときに、任意のタイマー値でタイマーを起動し、タイマー値を超えた時点で他のどのポートからもParent Notify信号を受信できないときそのポートの先に接続されるデバイス間で構成されるバス上でループが存在すると判断する。

【0049】また好ましくは、前記ループ検知工程で用いられる任意のタイマー値と、前記判断工程で用いられる任意のタイマー値とは、別個の値が設定され、前記ループ検知工程と前記判断工程の起動タイミングを変えていることを特徴とする。

【0050】また好ましくは、前記ディセーブル工程によってポートがディセーブルにされた場合、どのポートがディセーブルにされたかユーザに通知する通知工程を更に備える。

【0051】また、コンピュータ可読の記録媒体はデバイスの追加接続や切離しを検知しバスの構成を行うバス構成工程と、バス構成時インタフェースシステム上におけるループ接続されたデバイスの存在を検知するループ検知工程と、他のデバイスとの接続口となるポートから強制的にParent Notify信号を発生させる信号発生工程と、前記ポートからParent Notify信号を受信する信号受信工程と、前記信号の送信および受信状況からループ接続の発生箇所を判断する判断工程と、前記判断工程の判断結果に基づきアクティブなポートをディセーブルにするディセーブル工程と、前記ディセーブル工程によるポートのディセーブル後に前記バス構成工程を再び開始

させるシステム再構成開始工程とを備える。

【0052】また、複数のデバイスがシリアルバスインタフェースデバイスで接続されて構成されるシリアルバスインタフェースシステムは、前記シリアルバスインタフェースデバイスとして、前記シリアルバスインタフェースデバイスを用いたことを特徴とする。

【0053】また好ましくは、前記シリアルバスインタフェースデバイスで接続されたデバイスはプリンタ機能を備える。

10 【0054】この構成によって、確実なループ検知を行いそのループ接続をユーザの手を煩わせることなく自動的に回避してバスを再構成する。また再構成できない場合も含めてループ接続の発生箇所をユーザに対して的確に知らせることができる。

【0055】

【発明の実施の形態】〔第1の実施形態〕本発明の第1の実施形態を図に従って説明する。

【0056】図1は、第1の実施形態におけるデバイス内の構成を示すブロック図である。

20 【0057】101は、IEEE1394の規格に準拠したインタフェースを備えるノード（デバイス本体）である。以下に説明する102、105～113の各ブロックで構成される。この例では画像読み取りを行うスキャナデバイスを想定している。この他にもIEEE1394インタフェースシステムを構成できるデバイスがあれば上位レイヤであるアプリケーションの機能としては何でも構わない。

【0058】102は、IEEE1394インタフェースのデバイス本体側の受け口（ポート）となるリセプタクルである。ここにシステムを構成する各種デバイスとの接続をするためのケーブルが差し込まれる。

【0059】103は、IEEE1394インタフェースケーブルのプラグである。この部分を各種デバイスのポートにはめ込んで接続する。

【0060】104は、IEEE1394インタフェースのケーブルである。このケーブル内には2組のツイストペアケーブル（一方がA、他方がBと称される信号線）と1組の電源ペアケーブルのあわせて6本のケーブルがクロスしている。

【0061】105は、IEEE1394に準拠した100Mbps～400Mbpsの転送スピードをサポートする物理レイヤを実現するブロックで各種のハードウェアロジック（ツイステッドペアのインタフェースの複数ポート、リンクレイヤICへのインタフェース、パケットデータのシンクロナイズと再構成、ビットレベルのアービトラーション、および本発明の要点となる初期化ロジック、タイマー等）によって構成される。機能および構成の詳細については本発明の骨子となるものであるが、構成に関しては図2を、その処理フローに関しては図3～6を用いて後ほど詳述する。

50 【0062】106は、IEEE1394に準拠したリンクレイ

ヤのコントローラで各種ハードウェアロジックによって構成される。

【0063】107は、本デバイスの各種制御およびIEEE1394インタフェースのトランザクションレイヤ、ノードコントローラおよびアプリケーションレイヤの機能を実現するCPUである。

【0064】108は、CPU107が各種処理を行う際に必要な命令およびデータを格納するRAMである。IEEE1394インタフェースによる転送データの記憶やスキヤナによって読み込まれた画像の一時記憶にも使用される。

【0065】109は、本デバイスの制御情報を格納してあるROMである。フラッシュメモリなどを利用することで制御情報の更新を後から行うことも可能である。

【0066】110は、本デバイスのメカ制御および各種画像処理を行うASICである。画像読み込み時のスキヤナヘッドの制御や読み込んだ画像データの補正などの各種画像処理を行う専用のハードウェアロジックである。

【0067】111は、スキヤナである。原稿読み取り時の光源、CCDセンサ、A/D変換器、画信号補正回路などによって構成される。

【0068】112は、ユーザーへステータスを通知したり、ユーザーからのコマンド入力を受け付けるためのユーザーインタフェースである。通知するための表示部と入力を受け付ける操作部によって構成される。

【0069】113は、システムバスである。図示されている106～112の各ブロックの他、不図示の構成要素もこのバス上にぶら下がっており、各ブロック間で高速なデータ転送を行うことができる。

【0070】図2は、第1の実施形態におけるPHYブロックの構成図である。

【0071】201は、図1の105の部分にあたる物理レイヤを構成するハードウェアロジックである。以下に説明する207～208、212～214のブロックで構成される。

【0072】202a、202bおよび202cは、IEEE1394インタフェースのデバイス本体側の受け口（ポート）となるリセプタクルである。ここに他の各種デバイスとの接続をするためのケーブルが差し込まれる。この実施形態ではデバイスは3つのポートを持つものとする。

【0073】203は、IEEE1394インタフェースケーブルのプラグである。この部分を各種デバイスのポートにはめ込んで接続する。

【0074】204は、IEEE1394インタフェースのケーブルである。このケーブル内には2組のツイストペアケーブル（一方がA、他方がBと称される信号線）と1組の電源ペアケーブルのあわせて6本のケーブルがクロスしている。

【0075】205は、TpAレシーバからのTpA信号である。IEEE1394デバイスはポート一つに対してTpAとTpBという二つのトランシーバを持っている。TpAは、アービトレーションビットとパケット（ストロブ信号）を送信し、アービトレーション信号とパケット（データ信号）を受信する。

【0076】206は、TpBレシーバからのTpB信号である。TpBは、アービトレーションビットとパケット（データ信号）を送信し、アービトレーション信号とパケット（ストロブ信号）を受信する。

【0077】207a、207bおよび207cは、差動信号を駆動するドライバおよび低オフセットの広帯域レシーバおよびディジタルの信号への変換部で構成されるトランスミッタおよびレシーバである。ここでIEEE1394のケーブル上を送信されてきた信号を受信したり、本デバイスから送信されるデータをケーブルに流す信号に変換する。

【0078】208は、クリスタルオシレータである。

【0079】209は、クリスタルオシレータ208の出力からIEEE1394インタフェースの動作周波数のクロックを生成するPLLである。

【0080】210は、IEEE1394インタフェースの動作クロックである。デバイスによって100Mbpsから400Mbpsの動作クロックが必要となる。

【0081】211は、トランスミッタおよびレシーバ207で変換されたディジタルの信号である。バスの構成時に使用される各種信号や送受信されるデータ信号がそれに相当する。

【0082】212は、リンクレイヤとのインタフェースの制御、バス初期化時、ループ検知時および再構成時の各種制御等を行う制御部（コントロールユニット）である。ループ検知時に使用するタイマーも含まれる。制御部での処理の詳細については後述する。

【0083】213は、IEEE1394インタフェースの仕様で定められているDS-Link方式の符号復号器である。

【0084】214は、リンクレイヤとのインタフェースであり、各種レジスタによって構成される。このインタフェースを通して受信したデータや他のデバイスに送信するデータを送る。またループ接続の存在や発生箇所の情報などを上位レイヤに伝える際にも使用される。

【0085】図3は、第1の実施形態における制御手順の概略であり、その中でもバスリセット時からIEEE1394インタフェースシステムというアプリケーションレイヤが動作できるところまでのフローを示している。

【0086】ステップ301では、バスのリセットが実行されてからループ検知処理に入るまでの初期化の処理を行う。詳細については図4を用いて後ほど説明する。

【0087】ステップ302では、初期化終了後イレギュラーな接続形態であるループ接続があるかどうか、またその接続形態がバス上のどの場所に検知されたのかを

判断するループ検知処理を行う。詳細については図5を用いて後ほど説明する。

【0088】ステップ303では、ループ接続を検知したかどうかの判断を行う。ループ接続を検知した場合はステップ306へ、ループ接続を検知せず正常なバスを構成できたと判断したときはステップ304へすすむ。

【0089】ステップ304では、正常なバスの構成だけが行われたと判断したのを受けてツリー識別を行う。ツリー識別の詳細については従来例の説明でしたので参照のこと。

【0090】ステップ305では、ツリー識別が行われたのを受けて自己識別を行う。具体的には各種ノードを含んだセルフIDパケットの送出および他のノード（デバイス）からのセルフIDパケットを受信することでバス上にいくつのデバイスが接続されているかなどの情報を得て、バスリセットのプロセスを終了する。

【0091】ステップ306では、ループ接続を検知したのを受けて、自ノードのアクティブなポートのいずれかをディセーブルにすることでバスの正常な構成が可能かどうかで処理を分岐する。分岐の判断自体はステップ302のループ検知処理で行う。バスの再構成が可能な場合はステップ307へ、バスの再構成ができない場合はステップ308へそれぞれすすむ。

【0092】ステップ307では、バスの再構成が可能だという判断を受けてバスの再構成を行う。バスの再構成の詳細については図6を用いて後ほど説明する。

【0093】ステップ308では、バスの再構成ができないことを受けてループ接続を検知した箇所をユーザーに通知する。通知する手段はデバイスにより様々であるが、表示装置への表示、紙などのメディアに対する結果の出力、音声によるものなどが考えられる。

【0094】図4は、第1の実施形態における初期化フローである。

【0095】本動作フローは初期化プロセスを実行するフローチャートを示しており、デバイスが接続されているポートからのバスリセット信号を検知したときおよびバスリセットを自らのノードで発生したときからスタートし、初期化プロセスが完了した時点で終了となる。

【0096】ステップ401では、バスリセットの開始を受けて自ノードにアクティブなポートが存在するかどうかを検出する。アクティブなポートがない場合はステップ402へ、ある場合はステップ404へすすむ。

【0097】ステップ402では、アクティブなポートが存在しない、つまりこのノードに対して接続されているデバイスがないことからスタンドアロンのデバイスであると判断し、スタンドアロンのデバイスとしての動作を開始する。

【0098】ステップ403では、スタンドアロンデバイス動作の開始を受け、このループ検知の処理から抜ける。

【0099】ステップ404では、バスリセットを発生させるための送信要求が上位レイヤからあるかどうかを判断する。バスリセットの送信要求の場合はステップ405へ、そうでない場合はステップ408へすすむ。

【0100】ステップ405では、バスリセットの送信要求を受けて、アクティブなポートすべてに対してバスリセット信号を送信する。

【0101】ステップ406では、バスリセット信号を送信したアクティブなポートすべてにアイドル信号を送信する。

【0102】ステップ407では、アクティブなポートすべてからアイドル信号を受信したかどうかを判断する。すべてのポートからアイドル信号を受信するまでこのルーチンを繰り返し、すべて受信した時点で初期化ルーチンを終了する。

【0103】ステップ408では、バスリセットが送信要求ではないのを受けてバスリセット信号を受信する。

【0104】ステップ409では、他にアクティブなポートが存在するかどうかを確認する。ある場合はステップ409へ、ない場合はバスリセット後の初期化ルーチンを終了する。

【0105】ステップ410では、バスリセットを受けたポート以外のすべてのアクティブなポートに対してバスリセット信号を送信する。

【0106】ステップ411では、ステップ410に続いてバスリセット信号を送信したすべてのアクティブなポートに対してアイドル信号を送信する。

【0107】ステップ412では、アクティブなポートすべてからアイドル信号を受信したかどうかを判断する。すべてのポートからアイドル信号を受信するまでこのルーチンを繰り返し、すべて受信した時点で初期化ルーチンを終了する。

【0108】以上のステップを踏んで初期化ルーチンが行われる。

【0109】図5(a)～(e)は、第1の実施形態におけるループ検知フローである。

【0110】本動作フローはループ検知プロセスを実行するフローチャートを示しており、初期化プロセスが完了した時点からスタートし、ループ検知プロセスが完了した時点で終了となる。

【0111】まずは図5(a)を用いて説明する。

【0112】ステップ501では、ツリー識別がある時間内に行われたかどうかを判断するためにある任意のタイマー値を持つタイマーを起動する。ここでは仮にタイマー1とする。

【0113】ステップ502では、自ノードの持つアクティブなポートの数を確認する。アクティブなポートが1つかそれより多いかを判断し、アクティブなポート数が2つ以上の場合はAへ、アクティブなポートが1つの場合はステップ503へ移行する。Aへ移行した際の処

理の流れに関しては図5 (b) を用いて説明する。

【0114】ステップ503では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ504へ、受信しなかった場合はステップ505へすすむ。

【0115】ステップ504では、Parent Notify信号を受信したのを受けてそのノードに対してChild Notify信号を送信し、ステップ520へ移行する。

【0116】ステップ505では、Parent Notify信号を受信していないのを受けて自ノードからそのノードに対してParent Notify信号を送信する。

【0117】ステップ506では、タイマー1がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ509へ、タイムアウトした場合はステップ507へすすむ。

【0118】ステップ507では、タイマー1のタイムアウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0119】ステップ508では、接続先のノードが無応答であると認識したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知するか、もしくはツリー構成にそのデバイスを加えない等の処理を行う。

【0120】ステップ509では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ520へ、受信しなかった場合はステップ510へすすむ。

【0121】ステップ510では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ511へ進み、受信しなかった場合はステップ506へ戻る。

【0122】ステップ511では、タイマー1を停止した後にタイマー2を起動する。ここでタイマー2は、ルート競合のためのタイマーで、競合した2つのデバイスだけが次にParent Notify信号を送信するまでの時間を決めるための時間をランダムに選択するためのものである。

【0123】ステップ512では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ513へ、受信しなかった場合はステップ515へすすむ。

【0124】ステップ513では、Parent Notify信号を受信したのを受けて、Child Notify信号を送信する。

【0125】ステップ514では、競合する2つのデバイス間で信号のやりとりが行われたことから自ノードがルートデバイスに確定する。

【0126】ステップ515では、Parent Notify信号を受信していないのを受けて自ノードからそのノードに対してParent Notify信号を送信する。

【0127】ステップ516では、タイマー2がカウン

トしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ517へ、タイムアウトした場合はステップ518へすすむ。

【0128】ステップ517では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ520へ進み、受信しなかった場合はステップ516へ戻る。

【0129】ステップ518では、タイマー2のタイムアウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0130】ステップ519では、接続先の相手ノードが無応答であると判断したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知したり、ツリー構成にそのデバイスを加えない等の処理を行う。

【0131】ステップ520では、ツリー識別処理の終了を確認する。

【0132】図5 (b) を用いてAへ移行した際の処理の流れを説明する。

【0133】ステップ521では、アクティブなポート数をカウントしてその値を保存する。仮にここではNというパラメータにその値を格納する。

【0134】ステップ522では、アクティブなポート数Nが2つであるかどうかを判断する。アクティブなポートが2つの場合はステップ523へ、3つ以上のアクティブなポートを持つ場合はCへ移行する。Cへ移行した際の処理の流れに関しては図5 (d) を用いて説明する。

【0135】ステップ523では、タイマー1がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ524へ、タイムアウトした場合はステップ527へすすむ。

【0136】ステップ524では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ525へ進み、受信しなかった場合はステップ523へ戻る。

【0137】ステップ525では、Parent Notify信号を受信したのを受けてそのノードに対してChild Notify信号を送信する。

【0138】ステップ526では、Parent Notify信号を受信したポートではないもう一方のポートに対してParent Notify信号を送信する。その後処理はBへ移行するがその処理の流れに関しては図5 (c) を用いて説明する。

【0139】ステップ527では、タイマー1がタイムアウトしたのを受けてタイマー3を起動する。タイマー1のタイムアウトの時点で無応答の状態が続いているポートが存在するため第一の判断としてループが発生したと認識する。通常のループ接続の検知であればここで処理を終了してしまうが、本発明では次に正確なループ

接続の発生箇所の認識とループ回避のためのプロセスに入る。タイマー3は無応答なデバイスに対して強制的にParent Notify信号を送信した場合別のポートからParent Notify信号が帰ってくるまでのタイムアウトの時間をカウントするタイマーである。

【0140】ステップ528では、無応答なポートの何れかに対してParent Notify信号を送信する。

【0141】ステップ529では、タイマー3がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ532へ、タイムアウトした場合はステップ530へすすむ。

【0142】ステップ530では、タイマー3のタイムアウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0143】ステップ531では、接続先の相手ノードが無応答であると認識したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知したり、ツリー構成にそのデバイスを加えない等の処理を行う。

【0144】ステップ532では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ533へ進み、受信しなかった場合はステップ529へ戻りこの間のシーケンスを繰り返す。

【0145】ステップ533では、タイマー3がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ535へ、タイムアウトした場合はステップ534へすすむ。

【0146】ステップ534では、タイマー3のタイムアウトを受けてループ接続の箇所が自ノードから強制的にParent Notify信号を送信したノードもしくはその接続先のノードであると判断する。またその際Parent Notify信号を送信したポートをディセーブルにしてもバスの再構成ができないこともあわせて判断し、それらの情報を記憶、保存する。

【0147】ステップ535では、自ノードから強制的にParent Notify信号を送信したポートとは別のポートからParent Notify信号を受信したかどうかを判断する。受信した場合はステップ536へ進み、受信しなかった場合はステップ533へ戻りこの間のシーケンスを繰り返す。

【0148】ステップ536では、自ノードから強制的にParent Notify信号を送信したポートとは別のポートからParent Notify信号を受信したことを受けてループ接続の発生箇所が自ノードの2つのポートの間に接続されるバスの中に存在すると判断する。またあわせて自ノードのどちらかのポートをディセーブルにすることによりバスの再構成が可能であることも判断し、それらの情報を記憶、保存する。ステップ534および本ステップで記憶、保存した情報はバスの再構成時にアクティブなポートをディセーブルにする場合およびループ接続の発

生箇所をユーザーに通知する場合に使用される。

【0149】図5(c)を用いて「B」へ移行した際の処理の流れを説明する。

【0150】ステップ537では、タイマー1がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ540へ、タイムアウトした場合はステップ538へすすむ。

【0151】ステップ538では、タイマー1のタイムアウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0152】ステップ539では、接続先の相手ノードが無応答であると認識したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知するか、もしくはツリー構成にそのデバイスを加えない等の処理を行う。

【0153】ステップ540では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ551へ、受信しなかった場合はステップ541へすすむ。

【0154】ステップ541では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ542へ進み、受信しなかった場合はステップ537へ戻る。

【0155】ステップ542では、タイマー1を停止した後にタイマー2を起動する。ここでタイマー2は、ルート競合のためのタイマーで、競合した2つのデバイスだけが次にParent Notify信号を送信するまでの時間を決めるための時間をランダムに選択するためのものであることは以前説明したとおりである。

【0156】ステップ543では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ544へ、受信しなかった場合はステップ546へすすむ。

【0157】ステップ544では、Parent Notify信号を受信したのを受けて、Child Notify信号を送信する。

【0158】ステップ545では、競合する2つのデバイス間で信号のやりとりが行われたことから自ノードがルートデバイスに確定する。

【0159】ステップ546では、Parent Notify信号を受信していないのを受けて自ノードからそのノードに対してParent Notify信号を送信する。

【0160】ステップ547では、タイマー2がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ548へ、タイムアウトした場合はステップ549へすすむ。

【0161】ステップ548では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ551へ進み、受信しなかった場合はステップ547へ戻る。

【0162】ステップ549では、タイマー2のタイム

アウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0163】ステップ550では、接続先の相手ノードが無応答であると認識したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知したり、ツリー構成にそのデバイスを加えない等の処理を行う。

【0164】ステップ551では、ツリー識別処理の終了を認識する。

【0165】図5（d）を用いてCへ移行した際の処理 10の流れを説明する。

【0166】ステップ552では、タイマー1がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ553へ、タイムアウトした場合はステップ558ですすむ。

【0167】ステップ553では、Parent Notify信号を受信したかどうかを判断する。受信した場合はステップ554へ進み、受信しなかった場合はステップ552へ戻る。

【0168】ステップ554では、Parent Notify信号 20を受信したのを受けて、Child Notify信号を送信する。

【0169】ステップ555では、アクティブなポート数Nの値から1をマイナスし、その値をパラメータNに格納する。

【0170】ステップ556では、アクティブなポート数Nの値が1かどうかを判断する。1の場合はステップ557へ進み、それ以外の場合つまりParent Notify信号を受信していないアクティブなポートが1つになるまではステップ552へ戻りこの間のシーケンスを繰り返す。

【0171】ステップ557では、アクティブなポートが残り一つになったのを受けてParent Notify信号を残されたポートに対して送信する。その後処理はBへ移行するがBへ移行後のフローについては図5（c）で説明したのでそちらを参照のこと。

【0172】ステップ558では、タイマー1がタイムアウトしたのを受けてタイマー3を起動する。タイマー1のタイムアウトの時点で無応答の状態が続いているポートが存在するため第一の判断としてループが発生したと認識する。タイマー3は無応答なデバイスに対して強制的にParent Notify信号を送信した場合別のポートからParent Notify信号が帰ってくるまでのタイムアウトの時間をカウントするタイマーである。

【0173】ステップ559では、無応答なポートの何れかに対してParent Notify信号を送信する。また送信したポートを記憶しておきこの処理よりを一度経過した後再びこのステップの処理を行う場合には一度送信したポートとは別のポートに対してParent Notify信号を送信する。

【0174】ステップ560では、タイマー3がカウ 50

トしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ563へ、タイムアウトした場合はステップ561へすすむ。

【0175】ステップ561では、タイマー3のタイムアウトを受けて相手ノード（デバイス）が無応答であると認識する。

【0176】ステップ562では、接続先の相手ノードが無応答であると認識したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知したり、ツリー構成にそのデバイスを加えない等の処理を行い、ループ検知処理のプロセスを終了する。

【0177】ステップ563では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ564へ進み、受信しなかった場合はステップ560へ戻りこの間のシーケンスを繰り返す。

【0178】ステップ564では、タイマー3がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ565へ、タイムアウトした場合はステップ568へすすむ。

【0179】ステップ565では、自ノードから強制的にParent Notify信号を送信したポートとは別のポートからParent Notify信号を受信したかどうかを判断する。受信した場合はステップ566へ進み、受信しなかった場合はステップ564へ戻りこの間のシーケンスを繰り返す。

【0180】ステップ566では、自ノードから強制的にParent Notify信号を送信したポートとは別のポートからParent Notify信号を受信したことを受けてループ 30接続の発生箇所が自ノードの2つのポートの間に接続されるバスの中に存在すると判断する。あわせて自ノードのどちらからのポートをディセーブルにすることによりバスの再構成が可能であることも判断し、それらの情報を記憶、保存する。

【0181】ステップ567では、アクティブなポート数Nの値から現在ループ接続箇所を判断した2つのポートを除くため2をマイナスしさらに残されたポート数がいくつかを判断する。残されたポート数が2以上の場合はステップ559へ戻りこの間の処理シーケンスを繰り返す。そうでない場合つまり残されたポートが1つの場合は図5（a）のステップ503まで戻り、アクティブポートが1つの場合の処理を行う。

【0182】ステップ568では、タイマー3のタイムアウトを受けてループ接続の発生箇所が自ノードから強制的にParent Notify 信号を送信したノードもしくはその接続先のノードによって構成されるバス上にあると判断する。またその際Parent Notify信号を送信したポートをディセーブルにしてもバスの再構成ができないこともあわせて判断する。

【0183】ステップ569では、ステップ568で判

断した内容を記憶保持し、アクティブなポート数Nの値から1マイナスし、その値をパラメータNに保存する。

【0184】ステップ570では、ステップ569までの各種情報を保持したままバスリセットを実行する。

【0185】ステップ571では、初期化のプロセスを行う。処理フローについては図4で既に説明しているのでそちらを参照のこと。

【0186】ステップ572では、ツリー識別がある時間内に行われたかどうかを判断するためのタイマーであるタイマー1を起動する。

【0187】ステップ573では、アクティブなポート数Nの値が0かどうかを判断する。Nが0でない場合、つまりParent Notify信号を送信することによってそのポートの先のバスの状況を把握していないポートがまだ存在している場合は、ステップ552へ戻りこの間のシーケンスを繰り返す。Nが0の場合はEへ移行する。Eへ移行した際の処理の流れに関しては図5(e)を用いて説明する。

【0188】図5(e)を用いてEへ移行した際の処理の流れを説明する。

【0189】この図では、複数あるアクティブなポートすべてがそれぞれそのポートを独立にディセーブルにすることでバスの再構成が不可能だと判断された場合の処理フローを示している。

【0190】ステップ574では、タイマー1がカウントしているタイマー値を超えていないかどうかを判断する。タイムアウトした場合はステップ575へ進み、超えていない場合は再びステップ574へ戻りこのステップを繰り返す。

【0191】ステップ575では、タイマー1がタイムアウトしたのを受けてタイマー3を起動する。タイマー1のタイムアウトの時点で無応答の状態が続いているポートが存在するため第一の判断としてループが発生したと認識する。タイマー3は無応答なデバイスに対して強制的にParent Notify信号を送信した場合別のポートからParent Notify信号が帰ってくるまでのタイムアウトの時間をカウントするタイマーである。

【0192】ステップ576では、Parent Notify信号を送信する複数のポートを選択する。仮に3つのポートA、B、Cがアクティブの場合にはこの中から二つのポートを選択する。また送信したポートを記憶しておき、この処理を一度経過した後再びこのステップの処理を行う場合には一度送信したポートとは別の組み合わせで複数のポートに対してParent Notify信号を送信する。組み合わせの際はアクティブなポート数Nよりも1小さい値まで同時にParent Notify信号を送信することが可能である。

【0193】ステップ577では、ステップ576で選択した複数のポートに対してParentNotify信号を送信する。

【0194】ステップ578では、タイマー3がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ581へ、タイムアウトした場合はステップ579へすすむ。

【0195】ステップ579では、タイマー3のタイムアウトを受けて相手ノードデバイスが無応答であると認識する。

【0196】ステップ580では、接続先の相手ノードが無応答であると判断したことによりエラー処理のシーケンスに入る。具体的には接続先の相手ノードが無応答であることをユーザーに通知したり、ツリー構成にそのデバイスを加えない等の処理を行い、ループ検知処理のプロセスを終了する。

【0197】ステップ581では、Child Notify信号を受信したかどうかを判断する。受信した場合はステップ582へ進み、受信しなかった場合はステップ578へ戻りこの間のシーケンスを繰り返す。

【0198】ステップ582では、タイマー3がカウントしているタイマー値を超えていないかどうかを判断する。超えていない場合はステップ583へ進み、タイムアウトした場合はステップ576へ戻り再びParent Notify信号を送信するポートを選択する。

【0199】ステップ583では、自ノードから強制的にParent Notify信号を送信した複数のポートとは別のポートからParent Notify信号を受信したかどうかを判断する。受信した場合はステップ584へ進み、受信しなかった場合はステップ582へ戻りこの間のシーケンスを繰り返す。

【0200】ステップ584では、自ノードから強制的にParent Notify信号を送信した複数のポートとは別のポートからParent Notify信号を受信したことを受けて、ループ接続の箇所が複数存在するという情報の他に、自ノードから強制的にParentNotify信号を送信した複数のポートをディセーブルにすることによりバスの再構成が可能であることも判断し、それらの情報を記憶、保存する。

【0201】ステップ585では、ステップ576で選択したポートの組み合わせが考え得るすべての組み合わせを選択したかどうかを判断する。すべて確認した場合はループ検知の処理プロセスを終了する。そうでない場合はステップ586へすすむ。また、すべての組み合わせを確認していない場合でもバスの再構成が可能な組み合わせを見つけているときはループ検知の処理プロセスを終了する。

【0202】ステップ586では、ステップ585までの各種情報を保持したままバスリセットを実行する。

【0203】ステップ587では、初期化のプロセスを行う。処理フローについては図4で既に説明しているの

【0204】ステップ588では、ツリー識別がある時

間内に行われたかどうかを判断するためのタイマーであるタイマー1を起動する。その後再びこの処理の最初であるステップ574から処理を行う。

【0205】以上のステップを踏んでループ検知プロセスが行われる。

【0206】図6は、第1の実施形態におけるバス再構成フローである。

【0207】ステップ601では、ループ接続の検知箇所をユーザーに通知するかどうか判断する。この判断は前もってユーザーが通知するかどうか判断し命令をこのデバイスに対して入力することによって行われる。通知する場合はステップ602へ、通知しない場合はステップ603へすすむ。

【0208】ステップ602では、ループ接続箇所の通知を行う。通知する手段はデバイスにより様々であるが、この実施形態ではユーザーインタフェースを構成する機能の一つである表示装置に対して表示を行うことで通知する。

【0209】ステップ603では、これ以前の処理の中で保存された情報に基づきアクティブなポートをディセーブルにする。

【0210】ステップ604では、ポートのディセーブル箇所をユーザーに通知するかどうか判断する。この判断は前もってユーザーが通知するかどうか判断し命令をこのデバイスに対して入力することによって行われる。通知する場合はステップ605へ、通知しない場合はステップ606へすすむ。

【0211】ステップ605では、ポートのディセーブル箇所の通知を行う。通知する手段は様々であるがこの実施形態ではユーザーインタフェースを構成する機能の一つである表示装置に対して表示を行うか、ポートがアクティブかどうかを表示するLEDなどを使用することで通知する。

【0212】ステップ606では、バスリセットを行い、バスの再構成プロセスを終了する。

【0213】以上のステップを踏んでバス再構成プロセスが行われる。

【0214】図7(a)～(g)は、第1の実施形態におけるループ接続を含む構成の模式図である。(以後第1のループ構成図という。)これらの図に従ってループ接続の検知してからバスを再構成するまでの流れを説明する。

【0215】図7(a)はループ接続を含むバスの構成図である。

【0216】ここで701～708はノードであり、ノード701が本発明の構成を備えるデバイスである。他は一つもしくは複数の接続ポートを備えたIEEE1394仕様に準拠した各デバイスである。

【0217】図のような接続がされたのちバスリセットが発生すると初期化のプロセスを経てツリー識別のフェ

ーズへ移行する。まず、Leafであるノード705、706、707および708からParent Notify信号がそれぞれ接続されたノードに対して送信される。Parent Notify信号を受信したBranchのノードは受信したポートにChild Notify信号を送信する。

【0218】しかしながら、ディジーチェーン接続やブランチ接続以外の許容されていない接続形態であるループ接続がノード701、702、703および704との間で成立しているため、この後ツリー識別を完了することができなくなることは従来例のところで説明した。

【0219】この状態からノード701がループ接続の箇所を判断し、バスを再構成する手順を以後の図によって説明する。

【0220】図7(b)は、図7(a)のループ接続を発生させているノードをピックアップしたものである。

【0221】図に示されるようにノード701から704までの各ノードはそれぞれアクティブなポートを2つ持つBranchとなるノードである。ノード701が本発明の構成を備えたデバイスである。

【0222】図7(c)では、ノード701はノード702に対してParent Notify信号を送信する。これはノード701が、ツリー識別のプロセスに移行した時点でタイマーを動作させ、ある任意のタイマー値を超えた時点でアクティブな2つのポートのどちらからもParent Notify信号を受信していないことからループ接続がどこかで発生していると判断し、その接続箇所とバスの再構成が可能であるかどうかを判断するためアクティブな2つのポートの一方に対して強制的にParent Notify信号を送信したものである。この例ではParent Notify信号をノード702に対して送信しているが、もう一方のノード703に対して送信しても構わない。

【0223】図7(d)では、ノード701からのParent Notify信号を受信したノード702がノード701に対してChild Notify信号を送信する。またノード702は、アクティブなポートでParent Notify信号を受信していないポートが残りの一つになったのでそのポートに対して、つまりノード704に対してParent Notify信号を送信する。

【0224】図7(e)では、それを受けて、ノード702からのParent Notify信号を受信したノード704がノード702に対してChild Notify信号を送信し、ノード703に対してParent Notify信号を送信する。さらにノード703はノード704へChild Notify信号を送信した後ノード701に対してParent Notify信号を送信する。

【0225】以上の過程を経て、ノード701はParent Notify信号を強制的に送信したポートとParent Notify信号を受信したポートの間にループ接続が存在することを判断するとともに、2つのアクティブなポート(ノード702および703が接続されているポート)のうち

どちらか一方をディセーブルすることでバスを構成することが可能なことも判断する。

【0226】図7(f)は、以上の判断に基づきノード701がノード702と接続されているポートをディセーブルにした場合のバスの構成図である。ディセーブル後にバスリセットを発生させバスの再構成を行った結果、ノード701、702がそれぞれLeafとなり正常にバスの構成を行うことができる。図7(a)でみると、ノード703と704がルート候補となるバスの構成になる。

【0227】図7(g)は、(f)と同様な判断に基づきノード701がノード703と接続されているポートをディセーブルにした場合のバスの構成図である。ディセーブル後にバスリセットを発生させバスの再構成を行った結果、ノード701、703がそれぞれLeafとなり正常にバスの構成を行うことができる。図7(a)でみると、ノード702と703と704がルート候補となるバスの構成になる。

【0228】以上のツリー識別プロセスが正常に完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別のフェーズに入る。

【0229】図8は、上述した図7を用いて説明した第1の実施形態における第1のループ構成例のループ検知のフローを示したものである。

【0230】ステップ801では、バスリセットなどのバスの構成を行うトリガーが発生したことを受けて初期化のプロセスが行われる。この処理内容の詳細は図4を用いた説明を参照のこと。

【0231】ステップ802では、ツリー識別がある時間内に行われたかどうかを判断するために、任意のタイマー値(タイマー1)を持つタイマーを起動する。

【0232】ステップ803では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかったことを受けてバスの中でループ接続が存在していると判断する。

【0233】ステップ804では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断するために接続先ノードであるノード702に対して強制的にParent Notify信号を送信する。

【0234】ステップ805では、Parent Notify信号を受信したノード702がノード701に対してChild Notify信号を送信する。

【0235】ステップ806では、ノード702がノード704に対してParent Notify信号を送信する。またノード704はノード702に対してChild Notify信号を送信する。

【0236】ステップ807では、ノード704がノード703に対してParent Notify信号を送信する。またノード703はノード704に対してChild Notify信号

を送信する。

【0237】ステップ808では、ノード703がノード701(強制的にParent Notify信号を送信したデバイス)に対してParent Notify信号を送信する。

【0238】ステップ809では、2つのアクティブなポート間に接続されるデバイスによってバス上にループ接続が構成されていると判断する。

【0239】ステップ810では、ユーザーに対してループの発生箇所を通知する。本実施形態ではノード701が持つ表示装置へ表示することで伝える。

【0240】ステップ811では、アクティブな2つのポートのうち、ノード702と接続しているポートをディセーブルにする。

【0241】ステップ812では、ユーザーに対してポートをディセーブルにしたことを通知する。本実施形態ではノード701の各ポートにはアクティブになると点灯するLEDが備えられておりそのLEDを点灯を止めることによって知らせる。

【0242】ステップ813では、バスリセットを発生させる。

【0243】ステップ814では、ステップ801と同様に初期化のプロセスを行う。

【0244】ステップ815では、ツリー識別のプロセスを行う。バスリセット前の処理では、このツリー識別プロセスの実行中にループ接続を検知したためバスの構成を続けることができなかったが、ポートをディセーブルにすることで正常にツリー識別プロセスを完了することができる。

【0245】ステップ816では、ツリー識別が正常に完了したのを受けて、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別プロセスに入る。この自己識別プロセスの完了後各ノード間での通信を行うことが可能になる。

【0246】図9(a)、(b)は、第1の実施形態におけるループ接続を含むバス構成の模式図である。(以後第2のループ構成図という)これらの図に従ってループ接続の検知してからバスを再構成するか否かを判断するまでの流れを説明する。

【0247】図9(a)はループ接続を含むバスの構成図である。

【0248】ここで901~904はノードであり、ノード901が本発明の構成を備えるデバイスである。他は一つもしくは複数の接続ポートを備えたIEEE1394仕様に準拠した各デバイスである。

【0249】図のようにノード902、903および904の間でループが形成されている。

【0250】図9(b)では、ツリー識別のプロセスで、まずツリー識別がある時間内に行われたかどうかを判断するために、任意のタイマー値(タイマー1)でタイマーを起動した後、アクティブなポートを一つしか持

たないLeafであるノード901はその接続先であるノード902に対してParent Notify信号を送信する。Parent Notify信号を受信したBranchとなるノード902はノード901にChild Notify信号を送信する。

【0251】しかしながら、ノード902はアクティブな3つのポートのうち1つからしかParent Notify信号を受けていないのでそれ以後の処理を続けることができないためノード901の持つタイマー1はタイムアウトする。つまりツリー識別プロセスを正常に終了することはできないことになる。

【0252】このことからノード901は、Parent Notify信号を送信したポートの先にループ接続を行っているデバイスが存在することを判断するとともに、このアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。バスの再構成ができないためループ接続が発生していることをユーザーに対して通知して処理を終了する。

【0253】図10は、上述した図9を用いて説明した第1の実施形態における第2のループ構成例のループ検知フローを示したものである。

【0254】ステップ1001では、バスリセットなどのバスの構成を行うトリガーが発生したことを受けて初期化のプロセスが行われる。この処理内容の詳細は図4を用いた説明を参照のこと。

【0255】ステップ1002では、ツリー識別がある時間内に行われたかどうかを判断するため、任意のタイマー値（タイマー1）を持つタイマーを起動する。

【0256】ステップ1003では、ノード901は唯一の接続先であるノード902に対してParent Notify信号を送信する。

【0257】ステップ1004では、Parent Notify信号を受信したノード902はノード901に対してChild Notify信号を送信する。

【0258】ステップ1005では、ループの構成によりそれ以後の処理を続けることができないためタイマー1はタイムアウトする。

【0259】ステップ1006では、タイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかったことを受けてParent Notify信号を送信した唯一のアクティブなポートの先にループ接続を行っているデバイスが存在することを判断する。さらにこのアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。

【0260】ステップ1007では、ユーザーに対してループの発生箇所を通知する。本実施形態ではノード701が持つ表示装置へ表示することで伝える。またアクティブなポートをディセーブルにすることで正常なバスの構成をすることができないこともあわせて表示する。

【0261】図11(a)～(d)は、第1の実施形態におけるループ接続を含むバス構成の模式図である。

(以後第3のループ構成図という)これらの図に従って接続の検知してからバスを再構成するまでの流れを説明する。

【0262】図11(a)はループ接続を含むバスの構成図である。

【0263】ここで1101～1107はノードであり、ノード1101が本発明の構成を備えるデバイスである。他は一つもしくは複数の接続ポートを備えたIEEE 1394仕様に準拠した各デバイスである。1108a～1108cはノード1101のポートである。

【0264】図のようにノード1101、1102、1104および1103の間とノード1105、1106および1107の間で二つのループが形成されている。この状態ではすべてのノードがBranchとなりツリー識別のプロセスを正常に行えない状態にある。

【0265】図11(b)では、ノード1101はポート1108aに接続されているノード1102に対してParent Notify信号を送信する。これはノード1101がツリー識別のプロセスに移行した時点でタイマーを動作させ、ある任意のタイマー値（タイマー1）を超えた時点でアクティブな3つのポートの何れからもParent Notify信号を受信していないことからループ接続がどこかで発生していると判断し、その接続箇所とバスの再構成が可能であるかどうかを判断するためアクティブな3つのポートの一つに対して強制的にParent Notify信号を送信したものである。この例ではParent Notify信号をノード1102に対して送信しているが、別のノードである1103、1105に対して送信しても構わない。なお送信前に送信したParent Notify信号に対する応答が時間内に行われるかどうかを判断するための上記とは別のタイマー値（タイマー3）でタイマーを起動する。

【0266】図11(c)では、ノード1101からのParent Notify信号を受信したノード1102がノード1101に対してChild Notify信号を送信する。またノード1102はアクティブなポートでParent Notify信号を受信していないポートが残りの一つになったのでそのポートに対して、つまりノード1104に対してParent Notify信号を送信する。同様にそれを受けて、ノード1102からのParent Notify信号を受信したノード1104がノード1102に対してChild Notify信号を送信し、ノード1103に対してParent Notify信号を送信する。さらにノード1103はノード1104へChild Notify信号を送信した後ノード1101に対してParent Notify信号を送信する。

【0267】この時点で、ノード1101はParent Notify信号を強制的に送信したポート1108aとParent Notify信号を受信したポート1108bの間にループ接続が存在することを判断するとともに、この2つのアクティブなポートのうちどちらか一方をディセーブルにすることでバスを構成可能なことも判断する。

【0268】アクティブなポートはもう一つあるので、上記判断を終えた時点でノード1105が接続しているポート1108cに対してParent Notify信号を送信する。Parent Notify信号を受信したBranchとなるノード1105はノード1101にChild Notify信号を送信する。

【0269】しかしながら、ノード1105はアクティブな3つのポートのうち1つからしかParent Notify信号を受けていないのでそれ以後の処理を続けることができないためタイマーはタイムアウトする。つまりツリー識別プロセスを正常に終了することはできないことになる。この場合の処理の流れは図9を用いて説明したものと同様である。

【0270】このことからノード1101は、Parent Notify信号を送信したポート1108cの先にループ接続を行っているデバイスが存在することを判断するとともに、このアクティブなポートをディセーブルにしてもこの先に接続されるデバイスで構成されるバスを有効にすることができないことも判断する。ただしこのポート1108cをディセーブルにすることでノード1101

【0271】図11(d)は上記判断に基づき、ポート1108aおよび1108cのポートをディセーブルにした場合のバスの構成図である。ディセーブル後にバスリセットを発生させバスの再構成を行った結果、ノード1101、1102がそれぞれLeafノードとなり正常にバスの構成を行うことができる。ポート1108aの代わりにポート1108bをディセーブルにしても同様のバスの再構成をはかることができる。またノード1105～1107はバスの構成に加われなくなるためユーザーの定義次第ではノード1101～1104間でバスを再構成することなくループ接続の箇所を知らせるにとどめることも可能である。

【0272】以上のツリー識別プロセスが完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別に入ることになる。

【0273】図12は、上述した図11を用いて説明した第1の実施形態における第3のループ構成例のループ検知フローを示したものである。

【0274】ステップ1201では、バスリセットなどのバスの構成を行うトリガーが発生したことを受けて初期化のプロセスが行われる。この処理内容の詳細は図4を用いた説明を参照のこと。

【0275】ステップ1202では、ツリー識別がある時間内に行われたかどうかを判断するため、任意のタイマー値(タイマー1)でタイマーを起動する。

【0276】ステップ1203では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかったことを受けてバスの中でループ接続が存在して

いると判断する。

【0277】ステップ1204では、次のステップで送信するParent Notify信号に対する応答があるかどうかを判断するため、上述のタイマー値とは別のタイマー値(タイマー3)でタイマーを起動する。

【0278】ステップ1205では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断するために接続先ノードであるノード1102に対して強制的にParent Notify信号を送信する。

10 【0279】ステップ1206では、Parent Notify信号を受信したノード1102がノード1101に対してChild Notify信号を送信する。

【0280】ステップ1207では、ノード1102がノード1104に対してParent Notify信号を送信する。それを受けてノード1104はノード1102に対してChild Notify信号を送信する。

20 【0281】ステップ1208では、ノード1104がノード1103に対してParent Notify信号を送信する。それを受けてノード1103はノード1104に対してChild Notify信号を送信する。

【0282】ステップ1209では、ノード1103がノード1101(強制的にParent Notify信号を送信したデバイス)に対してParent Notify信号を送信する。

【0283】ステップ1210では、ノード1101が2つのアクティブなポート1108aと1108bの間に接続されるデバイスによってループ接続が構成されていると判断する。あわせてこの2つのアクティブなポートのうちどちらか一方をディセーブルにすることでバスを構成可能なことも判断する。

30 【0284】ステップ1211では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断するために残ったアクティブなポートである1108c(接続ノードは1105)に対して強制的にParent Notify信号を送信する。

【0285】ステップ1212では、Parent Notify信号を受信したノード1105がノード1101に対してChild Notify信号を送信する。

40 【0286】ステップ1213では、ノード1105～1107によるループの構成によりそれ以後の処理を続けることができないためノード1101の持つタイマー3はタイムアウトする。

【0287】ステップ1214では、タイマー3のタイムアウト、つまりある時間内でツリー識別が完了しなかったことを受けてParent Notify信号を送信したアクティブなポート1108cの先にループ接続を行っているデバイスが存在することを判断する。さらにこのアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。またこのポート1108cをディセーブルにすることでノード1101～1104のデバイスのみでバスの再構成が可能であることを

あわせて判断する。

【0288】ステップ1215では、ユーザーに対してループ接続の発生箇所を通知する。本実施形態ではノード1101が持つ表示装置へ表示することで伝える。ユーザーが、バスに接続されたすべてのノード間でのバスの再構成を望むのであれば、ループ接続の発生箇所を通知後処理を終了する。

【0289】ステップ1216では、アクティブな3つのポートのうち、ポート1108aと1108cをディセーブルにする。

【0290】ステップ1217では、ユーザーに対してポートをディセーブルにしたことを通知する。本実施形態ではノード1101の各ポートにはアクティブになると点灯するLEDが備えられておりそのLEDを点灯を止めることによって知らせる。

【0291】ステップ1218では、バスリセットを発生させる。

【0292】ステップ1219では、ステップ1201と同様に初期化のプロセスを行う。

【0293】ステップ1220では、ツリー識別のプロセスを行う。バスリセット前はこのツリー識別プロセスの実行中にループ接続を検知したためバスの構成を続けることができなかったが、ポートをディセーブルにすることで正常にツリー識別プロセスを完了することができる。バスを構成するノードはノード1101～1104である。

【0294】ステップ1221では、ツリー識別が正常に完了したのを受けて、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別プロセスに入る。この自己識別プロセスの完了後各ノード間での通信を行うことが可能になる。

【0295】図13(a)～(h)は、第1の実施形態におけるループ接続を含むバス構成の模式図である。

(以後第4のループ構成図という)これらの図に従ってループ接続の検知してからバスを再構成するまでの流れを説明する。

【0296】図13(a)はループ接続を含むバスの構成図である。

【0297】ここで1301～1304はノードであり、ノード1301が本発明の構成を備えるデバイスである。他は一つもしくは複数の接続ポートを備えたIEEE1394仕様に準拠した各デバイスである。1305a～1305cはノード1301のポートである。

【0298】図のようにノード1301、1302および1303の間、ノード1301、1303および1304の間およびノード1301、1302、1303および1304の間など複雑に複数のループが形成されている。この状態ではすべてのノードがBranchとなりツリー識別のプロセスを正常に行えない状態にある。

【0299】図13(b)では、ノード1301はポー

ト1305aに接続されているノード1304に対してParent Notify信号を送信する。これはノード1301がツリー識別のプロセスに移行した時点でタイマーを動作させ、ある任意のタイマー値(タイマー1)を超えた時点でアクティブな3つのポートの何れからもParent Notify信号を受信していないことからループ接続がどこかで発生していると判断し、その接続箇所とバスの再構成が可能であるかどうかを判断する為アクティブな3つのポートの一つに対して強制的にParent Notify信号を送信したものである。この例ではParent Notify信号をまず最初にノード1304に対して送信しているが、別の接続ノードである1302や1303に対して送信しても構わない。なお送信前に送信したParent Notify信号に対する応答があるかどうかを判断するための上記とは別のタイマー値(タイマー3)でタイマーを起動する。

【0300】図13(c)では、ノード1301からのParent Notify信号を受信したノード1304がノード1301に対してChild Notify信号を送信する。またノード1304はアクティブなポートでParent Notify信号を受信していないポートが残り一つになったのでそのポートに対して、つまりノード1303に対してParent Notify信号を送信する。

【0301】しかしながら、ノード1303はアクティブな3つのポートのうち1つからしかParent Notify信号を受けていないのでそれ以後の処理を続けることができないためタイマー3はタイムアウトする。つまりツリー識別プロセスを正常に終了することはできないことになる。

【0302】このことからノード1301は、Parent Notify信号を送信したポート1305aの先にループ接続を行っているデバイスが存在することを判断するとともに、このアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。そしてここまで行った処理の内容と判断した結果を保存する。

【0303】図13(d)では、ノード1301は上記処理のあとバスリセットを行い、再び強制的なParent Notify信号の送信を行う。今回送信するポートは1305bである。ノード1301がツリー識別のプロセスに移行した時点でタイマーを動作させ、ある任意のタイマー値(タイマー1)を超えた時点でアクティブな3つのポートの何れからもParent Notify信号を受信していないことからループ接続がどこかで発生していると判断し、その接続箇所とバスの再構成が可能であるかどうかを判断するためアクティブな3つのポートの一つに対して強制的にParent Notify信号を送信するところはポート1305aに対するときと同じである。なお同様に送信前に送信したParent Notify信号に対する応答があるかどうかを判断するため、別のタイマー値(タイマー3)でタイマーを起動する。

【0304】その後ノード1301からのParent Notify信号を受信したノード1302はノード1301に対してChild Notify信号を送信する。またノード1302はアクティブなポートでParent Notify信号を受信していないポートが残り一つになったのでそのポートに対して、つまりノード1303に対してParent Notify信号を送信する。

【0305】しかしながら、ノード1303はアクティブな3つのポートのうち1つからしかParent Notify信号を受けていないのでそれ以後の処理を続けることができないためタイマー3はタイムアウトする。つまりツリー識別プロセスを正常に終了することはできないことになる。

【0306】このことからノード1301は、ポート1305aのときと同様にポート1305bの先にループ接続を行っているデバイスが存在することを判断するとともに、このアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。そしてここまで行った処理の内容と判断した結果を保存する。

【0307】図13(e)では、ノード1301は上記処理のあと再びバスリセットを行い、再び強制的なParent Notify信号の送信を行う。今回送信するポートは1305cである。送信前に送信したParent Notify信号に対する応答があるかどうかを判断するためのタイマーを起動するのも同様である。

【0308】その後ノード1301からのParent Notify信号を受信したノード1303はノード1301に対してChild Notify信号を送信する。この時点で、ノード1303はアクティブな3つのポートのうち1つからしかParent Notify信号を受けていないのでそれ以後の処理を続けることができないためタイマー3はタイムアウトする。つまりツリー識別プロセスを正常に終了することはできないことになる。

【0309】このことからノード1301は、ポート1305aおよび1305bのときと同様にポート1305cの先にループ接続を行っているデバイスが存在することを判断するとともに、このアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。そしてここまで行った処理の内容と判断した結果を保存する。

【0310】図13(f)では、ノード1301は上記処理のあと再びバスリセットを行い、強制的なParent Notify信号の送信を複数のポートに対して行う。今回送信するポートは1305aと1305bである。送信前に送信したParent Notify信号に対する応答があるかどうかを判断するためのタイマーを起動するのも同様である。この例ではParent Notify信号を1305aと1305bという二つのポートに対して送信しているが、この組み合わせは別（例えば1305bと1305c等）

でも構わない。

【0311】図13(g)では、ノード1301からのParent Notify信号を受信したノード1302および1304がノード1301に対してChild Notify信号を送信する。さらにノード1302と1304はParent Notify信号を受信していないアクティブなポートが残り一つになったので残りポートに対してそれぞれParent Notify信号を送信する。送信先のノードは1303であり1302および1304に対してChild Notify信号を送信する。この時点で、ノード1303のアクティブな残りポートも1つになったのでノード1303は1301に対してParent Notify信号を送信する。

【0312】ここまでの処理により、ノード1301は複数のループ接続がバス上に存在することを判断するとともにそれぞれアクティブなポートを独立にディセーブルにすることでバスの再構成をすることができないことも判断する。さらにParent Notify信号を送信した二つのポート1305aおよび1305bをディセーブルにすることでバスを構成可能なことも判断する。

【0313】図13(h)は上記判断に基づき、ポート1305aおよび1305bのポートをディセーブルにした場合のバスの構成図である。二つのポートをディセーブル後にバスリセットを発生させバスの再構成を行った結果、ノード1301、1302および1304がそれぞれLeafノードとなり正常にバスの構成を行うことができる。

【0314】以上のツリー識別プロセスが完了すると、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別に入ることになる。

【0315】図14は、上述した図13を用いて説明した第1の実施形態における第4のループ構成例のループ検知フローを示したものである。

【0316】ステップ1401では、バスリセットなどのバスの構成を行うトリガーが発生したことを受けて初期化のプロセスが行われる。この処理内容の詳細は図4を用いた説明を参照のこと。

【0317】ステップ1402では、ツリー識別がある時間内に行われたかどうかを判断するため、任意のタイマー値（タイマー1）でタイマーを起動する。

【0318】ステップ1403では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかったことを受けてバスの中でループ接続が存在していると判断する。

【0319】ステップ1404では、次のステップで送信するParent Notify信号に対する応答があるかどうかを判断するため、上述したタイマーとは異なるタイマー値（タイマー3）でタイマーを起動する。

【0320】ステップ1405では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断す

るために接続先ノードであるノード1304に対して強制的にParent Notify信号を送信する。

【0321】ステップ1406では、Parent Notify信号を受信したノード1304がノード1301に対してChild Notify信号を送信する。

【0322】ステップ1407では、ループの構成によりそれ以後の処理を続けることができないためタイマー3はタイムアウトする。

【0323】ステップ1408では、タイマー3のタイムアウト、つまりある時間内でツリー識別が完了しなかつたことを受けてParent Notify信号を送信したアクティブなポート1305aの先にループ接続を行っているデバイスが存在することを判断する。さらにこのアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。それからそれまでの処理内容および判断結果を保存する。

【0324】ステップ1409では、バスリセットを行う。

【0325】ステップ1410では、ステップ1401と同じく初期化のプロセスが行われる。

【0326】ステップ1411では、ステップ1402と同じくタイマー1を起動する。

【0327】ステップ1412では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかつたことを受けてバスの中でループ接続が存在していると判断する。

【0328】ステップ1413では、ステップ1404と同じくタイマー3を起動する。

【0329】ステップ1414では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断するために接続先ノードであるノード1302に対して強制的にParent Notify信号を送信する。

【0330】ステップ1415では、Parent Notify信号を受信したノード1302がノード1301に対してChild Notify信号を送信する。

【0331】ステップ1416では、ループの構成によりそれ以後の処理を続けることができないためタイマー3はタイムアウトする。

【0332】ステップ1417では、ステップ1408と同じくタイマー3のタイムアウトを受けてParent Notify信号を送信したアクティブなポート1305bの先にループ接続を行っているデバイスが存在することを判断する。さらにこのアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。それからそれまでの処理内容および判断結果を保存する。

【0333】ステップ1418では、バスリセットを行う。

【0334】ステップ1419では、ステップ1401と同じく初期化のプロセスが行われる。

【0335】ステップ1420では、ステップ1402と同じくタイマー1を起動する。

【0336】ステップ1421では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかつたことを受けてバスの中でループ接続が存在していると判断する。

【0337】ステップ1422では、ステップ1404と同じくタイマー3を起動する。

【0338】ステップ1423では、ループ接続の存在箇所の確認およびバスの再構成が可能かどうかを判断するために接続先ノードであるノード1303に対して強制的にParent Notify信号を送信する。

【0339】ステップ1424では、Parent Notify信号を受信したノード1303がノード1301に対してChild Notify信号を送信する。

【0340】ステップ1425では、ループの構成によりそれ以後の処理を続けることができないためタイマー3はタイムアウトする。

【0341】ステップ1426では、ステップ1408と同じくタイマー3のタイムアウトを受けてParent Notify信号を送信したアクティブなポート1305bの先にループ接続を行っているデバイスが存在することを判断する。さらにこのアクティブなポートをディセーブルにしてもバスを再構成することができないことも判断する。それからそれまでの処理内容および判断結果を保存する。

【0342】ステップ1427では、バスリセットを行う。

【0343】ステップ1428では、ステップ1401と同じく初期化のプロセスが行われる。

【0344】ステップ1429では、ステップ1402と同じくタイマー1を起動する。

【0345】ステップ1430では、そのタイマー1のタイムアウト、つまりある時間内でツリー識別が完了しなかつたことを受けてバスの中でループ接続が存在していると判断する。

【0346】ステップ1431では、ステップ1404と同じくタイマー3を起動する。

【0347】ステップ1432では、ポート1305aと1305bの2つのポートに対して強制的にParent Notify信号を送信する。

【0348】ステップ1433では、Parent Notify信号を受信したノード1302および1304がノード1301に対してChild Notify信号を送信する。

【0349】ステップ1434では、ノード1302がノード1303に対してParent Notify信号を送信する。またそれを受けてノード1303はChild Notify信号を1302に対して送信する。

【0350】ステップ1435では、ノード1304がノード1303に対してParent Notify信号を送信す

る。またそれを受けてノード1303はChild Notify信号を1304に対して送信する。

【0351】ステップ1436では、ノード1303はノード1302および1304からのParent Notify信号を受信したのを受けてノード1301に対してParent Notify信号を送信する。

【0352】ステップ1437では、ステップ1432で強制的にParent Notify信号を送信したポート1305aおよび1305bをディセーブルにすることでバスの再構成が可能であることを判断する。

【0353】ステップ1438では、ユーザーに対してループの発生箇所を通知する。本実施形態ではノード1301が持つ表示装置へ表示することで伝える。

【0354】ステップ1439では、アクティブな3つのポートのうち、ポート1305aと1305bをディセーブルにする。

【0355】ステップ1440では、ユーザーに対してポートをディセーブルにしたことを通知する。本実施形態ではノード1101の各ポートにはアクティブになると点灯するLEDが備えられておりそのLEDを点灯を止めることによって知らせる。

【0356】ステップ1441では、バスリセットを発生させる。

【0357】ステップ1442では、ステップ1401と同様に初期化のプロセスを行う。

【0358】ステップ1443では、ツリー識別のプロセスを行う。最初のバス構成時はこのツリー識別プロセスの実行中にループ接続を検知したためバスの構成を続けることができなかったが、複数のポートをディセーブルにすることで正常にツリー識別プロセスを完了することができる。

【0359】ステップ1444では、ツリー識別が正常に完了したのを受けて、次に各ノードにノード番号を割り振りお互いがデータ通信を可能にするための自己識別プロセスに入る。この自己識別プロセスの完了後各ノード間での通信を行うことが可能になる。

【0360】このように、複数のループが存在する場合にもループを回避して正常なバス構成を形成することが可能になる。

【0361】以上説明してきたように、本発明の構成によって、確実なループ接続の検知を行うとともにそのループ接続を自動的に回避してバスを再構成することができる。また再構成できない場合も含めてループ接続の発生箇所をユーザーに対して的確に知らせることができる。

【0362】〔第2の実施形態〕本発明の第2の実施形態を図にしたがって説明する。

【0363】本実施形態はユーザーへの通知手段を除いて実施形態1と同じ構成のため、同じ部分の説明は省略する。

【0364】図15は、第2の実施形態におけるデバイス内の構成を示すブロック図である。

【0365】1501は、IEEE1394の規格に準拠したインタフェースを備えるノード（デバイス本体）である。以下に説明する1502、1505～1513のブロックで構成される。この例では画像データの出力を行うプリンタデバイスを想定している。

【0366】1502は、IEEE1394インタフェースのデバイス本体側の受け口（ポート）となるレセプタクルである。ここに他の各種デバイスを接続するためのケーブルが差し込まれる。

【0367】1503は、IEEE1394インタフェースケーブルのプラグである。この部分を各種デバイスのポートにはめ込んで接続する。

【0368】1504は、IEEE1394インタフェースのケーブルである。このケーブル内には2組のツイストペアケーブル（一方がA、他方がBと称される信号線）と1組の電源ペアケーブルのあわせて6本のケーブルがクロスしている。

【0369】1505は、IEEE1394に準拠した100Mbps～400Mbpsの転送スピードをサポートする物理レイヤを実現するブロックで各種のハードウェアロジック（ツイステッドペアのインタフェースの複数ポート、リンクレイヤICへのインタフェース、パケットデータのシンクロナイズと再構成、ビットレベルのアビトラーション、および本発明の要点となる初期化ロジック、タイマー等）によって構成される。機能および構成の詳細については図2を参照のこと。

【0370】1506は、IEEE1394に準拠したリンクレイヤのコントローラで各種ハードウェアロジックによって構成される。

【0371】1507は、本デバイスの各種制御およびIEEE1394インタフェースのトランザクションレイヤ、ノードコントローラおよびアプリケーションレイヤの機能を実現するCPUである。

【0372】1508は、CPU1507が各種処理を行う際に必要な命令およびデータを格納するRAMである。IEEE1394インタフェースによる転送データの記憶やプリンタによって出力される画像データの一時記憶にも使用される。

【0373】1509は、本デバイスの制御情報を格納してあるROMである。フラッシュメモリなどを利用することで後から制御情報を更新することも可能である。

【0374】1510は、本デバイスのメカ制御および各種画像処理を行うASICである。画像データ印字時のプリンタヘッドの制御や送信された画像データの展開などの各種処理を行う専用のハードウェアロジックである。

【0375】1511は、プリンタである。メカ機構、インクタンク、インクヘッド等によって構成される。こ

の例ではインクジェットプリンタを想定しているが他に熱転写方式、電子写真方式など各種の方式を利用したプリンタの構成が考えられる。

【0376】1512は、ユーザーヘステータスを通知したり、ユーザーからのコマンド入力を受け付けるためのユーザーインタフェースである。通知するための表示部と入力を受け付ける操作部によって構成される。

【0377】1513は、システムバスである。図示されている1506～1512の各ブロックの他、不図示のブロックもこのバス上にぶら下がっており、各ブロック間での高速なデータ転送を行うことができる。

【0378】この構成では実施形態1と同じようにユーザーインタフェース1512の表示部にループ接続の発生箇所を知らせることができるだけでなく、デバイスの持つプリンタの機能によりメディア（一般には紙）に対してループ接続の発生箇所を印字して知らせることも可能である。またその場合発生箇所をテキスト形式の情報でプリントアウトするだけでなくツリー構造を図示してプリントすることも可能である。ループ接続の発生箇所を示すプリントデータの生成はプリンタ内のCPUで

行われる。

【0379】以上説明してきたように、本発明の構成によって、確実なループ接続の検知を行うとともにそのループ接続を自動的に回避してバスを再構成することができる。また再構成できない場合も含めてループ接続の発生箇所をユーザーに対して的確に知らせることができる。さらにこのデバイスがプリンタ機能を有するため表示装置への出力だけでなく紙などのメディアへの出力によってもループ接続の発生箇所をユーザーに知らせることができるため、表示装置を使用した通知よりもより直感的でより多くの情報を伝えることができる。

【0380】〔第3の実施形態〕本発明の第3の実施形態を図にしたがって説明する。

【0381】本実施形態はユーザーへの通知手段を除いて実施形態1と同じ構成のため、同じ部分の説明は省略する。

【0382】図16は、第3の実施形態におけるデバイス内の構成を示すブロック図である。

【0383】1601は、IEEE1394の規格に準拠したインタフェースを備えるノード（デバイス本体）である。以下に説明する1602、1605～1615のブロックで構成される。この例では動画像データの読み込み、記憶保存および出力を行うデジタルビデオカメラを想定している。

【0384】1602は、IEEE1394インタフェースのデバイス本体側の受け口（ポート）となるリセプタクルである。ここに他の各種デバイスを接続するためのケーブルが差し込まれる。

【0385】1603は、IEEE1394インタフェースケーブルのプラグである。この部分を各種デバイスのポート

にはめ込んで接続する。

【0386】1604は、IEEE1394インタフェースのケーブルである。このケーブル内には2組のツイストペアケーブル（一方がA、他方がBと称される信号線）と1組の電源ペアケーブルのあわせて6本のケーブルがクロスしている。

【0387】1605は、IEEE1394に準拠した100Mbps～400Mbpsの転送スピードをサポートする物理レイヤを実現するブロックで各種のハードウェアロジック（ツイステッドペアのインタフェースの複数ポート、リンクレイヤICへのインタフェース、パケットデータのシンクロナイズと再構成、ビットレベルのアービトレーション、および本発明の要点となる初期化ロジック、タイマー等）によって構成される。機能および構成の詳細については図2を参照のこと。

【0388】1606は、IEEE1394に準拠したリンクレイヤのコントローラで各種ハードウェアロジックによって構成される。

【0389】1607は、本デバイスの各種制御およびIEEE1394インタフェースのトランザクションレイヤ、ノードコントローラおよびアプリケーションレイヤの機能を実現するCPUである。

【0390】1608は、CPU1607が各種処理を行う際に必要な命令およびデータを格納するRAMである。IEEE1394インタフェースによる動画像データの転送や読み込んだ画像データのDVカセットへの書き込みをする際の一時記憶にも使用される。

【0391】1609は、本デバイスの制御情報を格納してあるROMである。フラッシュメモリなどを利用することで後から制御情報を更新することも可能である。

【0392】1610は、本デバイスのメカ制御および各種画像処理を行うASICである。レンズのオートフォーカスをはじめとするビデオカメラ部の各種制御、動画像データ保存時の磁気ヘッドの制御および送信する画像データの圧縮処理などの各種処理を行う専用のハードウェアロジックである。

【0393】1611は、ユーザーヘステータスを通知したり、ユーザーからのコマンド入力を受け付けるためのユーザーインタフェースである。通知するための表示部と入力を受け付ける操作部によって構成される。

【0394】1612は、動画像データを読み込む撮像部である。レンズの他各種補正回路等で構成される。

【0395】1613は、読み込んだ動画像データの読み込み保存を行う記録部である。DV仕様のカセットへの記録を行う。

【0396】1614は、不図示のマイクによって録音された音声信号を再生するスピーカである。

【0397】1615は、システムバスである。図示されている1606～1614の各ブロックの他、不図示のブロックもこのバス上にぶら下がっており、各ブロッ

ク間での高速なデータ転送を行うことができる。

【0398】この構成では実施形態1と同じようにユーザーインタフェース1611の表示部にループ接続の発生箇所を知らせることができるだけでなく、デバイスの持つスピーカ1614を使用して音声によるユーザーへの通知も可能である。

【0399】以上説明してきたように、本発明の構成によって、確実なループ接続の検知を行うとともにそのループ接続を自動的に回避してバスを再構成することができる。また再構成できない場合も含めてループ接続の発生箇所をユーザーに対して的確に知らせることができる。さらにこのデバイスがスピーカを有するため表示装置への出力だけでなく音声出力によってもループ接続の発生箇所をユーザーに知らせることでより注意を促すことができる。

【0400】

【他の実施形態】なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェース機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0401】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0402】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0403】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0404】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0405】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、そ

の処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0406】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図21のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。すなわち、少なくとも「バス構成モジュール2101」「ループ検知モジュール2102」および「信号発生モジュール2103」「信号受信モジュール2104」「判断モジュール2105」「ディセーブルモジュール2106」「システム再構成開始モジュール2107」の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【0407】

【発明の効果】以上説明したように、本発明によれば以下の効果をあげることができる。

【0408】確実なループ検知を行いそのループ接続を自動的に回避してバスを再構成する。また再構成できない場合も含めてループ接続の発生箇所をユーザーに対して的確に知らせることができる。

【0409】すなわち、確実なループ接続形態の検知によってバス構成時の信頼性が飛躍的に向上することになる。

【0410】また、ディスプレイ表示等によりループ接続の発生箇所をユーザーに対して的確に伝えることが可能になる。

【0411】本機能を備えるデバイスがループ接続を構成していなくてもシリアルバス上にループ接続が存在していればその発生箇所を認識することが可能になる。

【0412】ユーザーはケーブルを外す手間をかけることなく自動的にそのループ接続を回避して正常なバスを構成することが可能になる。つまりユーザーの利便性も飛躍的に向上する。

【0413】シリアルバス上に2つ以上のループ接続が発生してもその存在を検知することが可能になる。

【0414】

【図面の簡単な説明】

【図1】第1の実施形態におけるデバイス内の構成を示すブロック図である。

【図2】第1の実施形態におけるPHYブロックの構成図である。

【図3】第1の実施形態における制御手順の概略を示したフロー（初期化、ループ検知、バスの再構成）である。

【図4】第1の実施形態における初期化フローである。

【図5(a)】第1の実施形態におけるループ検知フローである。

【図5(b)】第1の実施形態におけるループ検知フローである。

【図5(c)】第1の実施形態におけるループ検知フロ

一である。

【図5(d)】第1の実施形態におけるループ検知フローである。

【図5(e)】第1の実施形態におけるループ検知フローである。

【図6】第1の実施形態におけるバス再構成フローである。

【図7(a)】第1の実施形態における第1のループ構成図である。

【図7(b)】第1の実施形態における第1のループ構成図である。

【図7(c)】第1の実施形態における第1のループ構成図である。

【図7(d)】第1の実施形態における第1のループ構成図である。

【図7(e)】第1の実施形態における第1のループ構成図である。

【図7(f)】第1の実施形態における第1のループ構成図である。

【図7(g)】第1の実施形態における第1のループ構成図である。

【図8】第1の実施形態における第1のループ構成の場合のループ検知フローである。

【図9(a)】第1の実施形態における第2のループ構成図である。

【図9(b)】第1の実施形態における第2のループ構成図である。

【図10】第1の実施形態における第2のループ構成の場合のループ検知フローである。

【図11(a)】第1の実施形態における第3のループ構成図である。

【図11(b)】第1の実施形態における第3のループ構成図である。

【図11(c)】第1の実施形態における第3のループ構成図である。

【図11(d)】第1の実施形態における第3のループ構成図である。

【図12】第1の実施形態における第3のループ構成の場合の処理フローである。

【図13(a)】第1の実施形態における第4のループ構成図である。

【図13(b)】第1の実施形態における第4のループ構成図である。

【図13(c)】第1の実施形態における第4のループ構成図である。

【図13(d)】第1の実施形態における第4のループ構成図である。

【図13(e)】第1の実施形態における第4のループ構成図である。

【図13(f)】第1の実施形態における第4のループ構成図である。

【図13(g)】第1の実施形態における第4のループ構成図である。

【図13(h)】第1の実施形態における第4のループ構成図である。

【図14(a)】第1の実施形態における第4のループ構成の場合の処理フローである。

【図14(b)】第1の実施形態における第4のループ構成の場合の処理フローである。

【図15】第2の実施形態におけるデバイス内の構成を示すブロック図である。

【図16】第3の実施形態におけるデバイス内の構成を示すブロック図である。

【図17】従来例におけるバスの構成図である。

【図18】従来例におけるツリーの識別(1)を示す図である。

【図19】従来例におけるツリーの識別(2)を示す図である。

【図20】従来例におけるループ接続を含むバスの構成図である。

【図21】記録媒体のメモリマップを示す図である。

【符号の説明】

101 ノード(デバイス本体)

102 リセプタクル

103 プラグ

104 ケーブル

105 PHY

106 LINK

107 CPU

108 RAM

109 ROM(フラッシュROM)

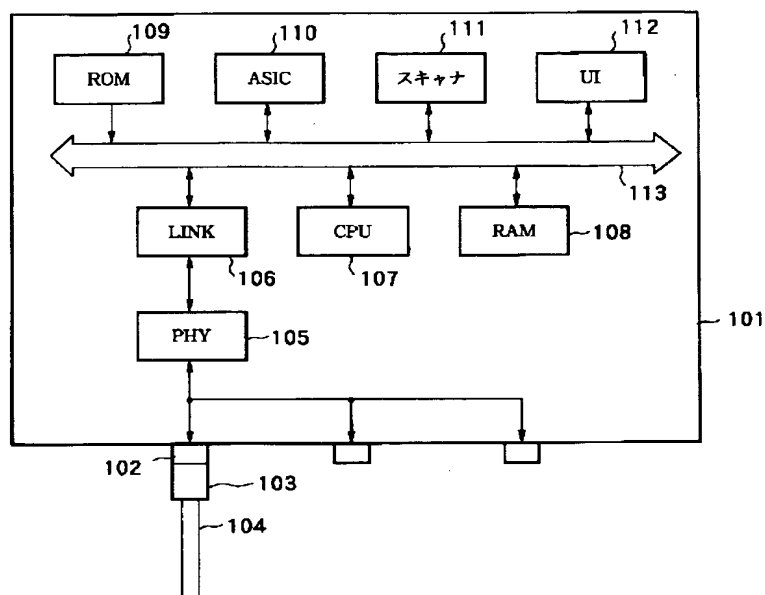
110 ASIC

111 スキャナ

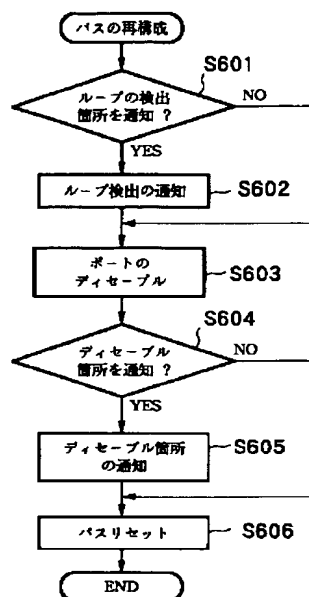
112 UI(ユーザーインタフェース)

113 システムバス

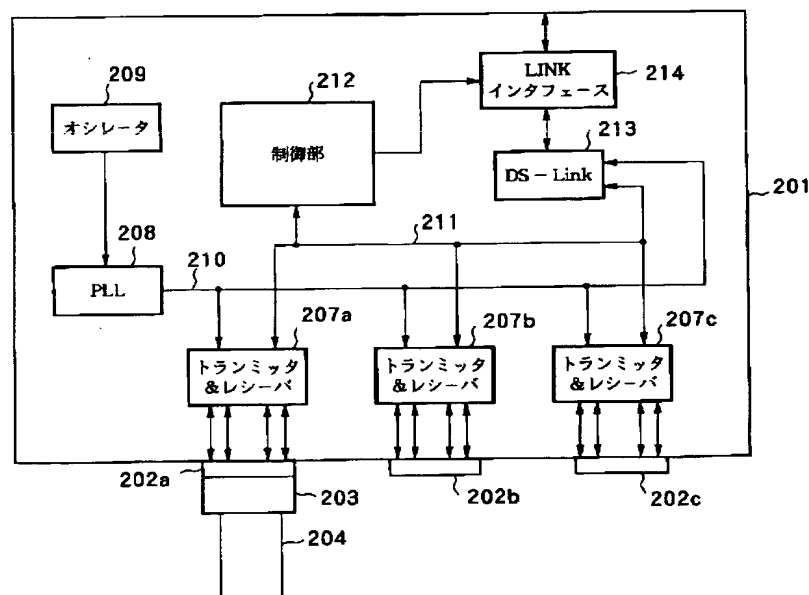
【図1】



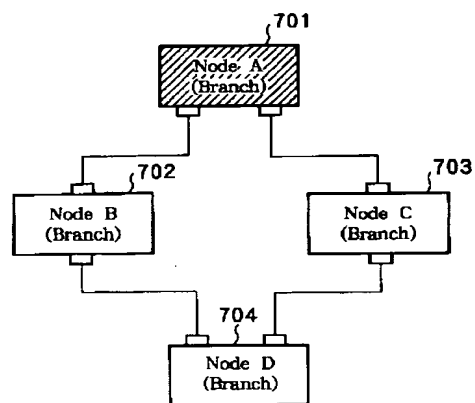
【図6】



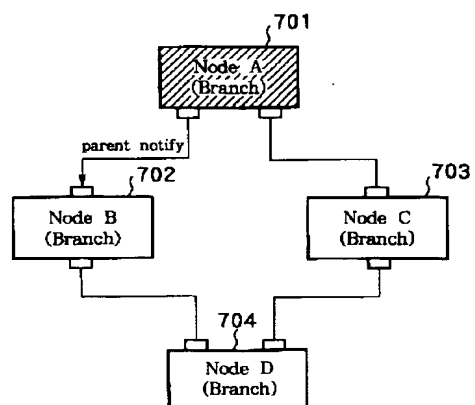
【図2】



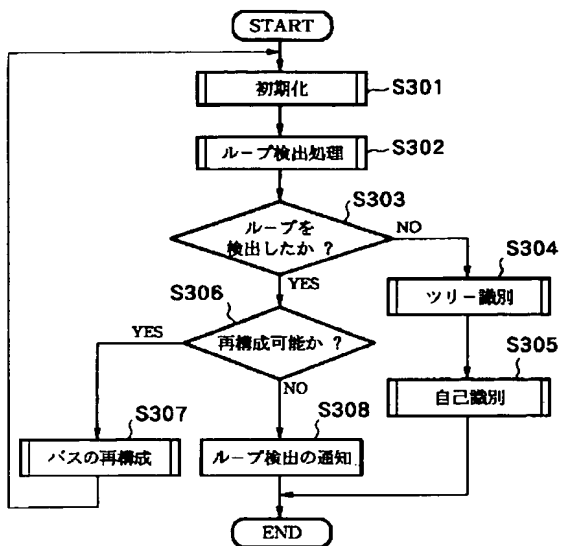
【図7 (b)】



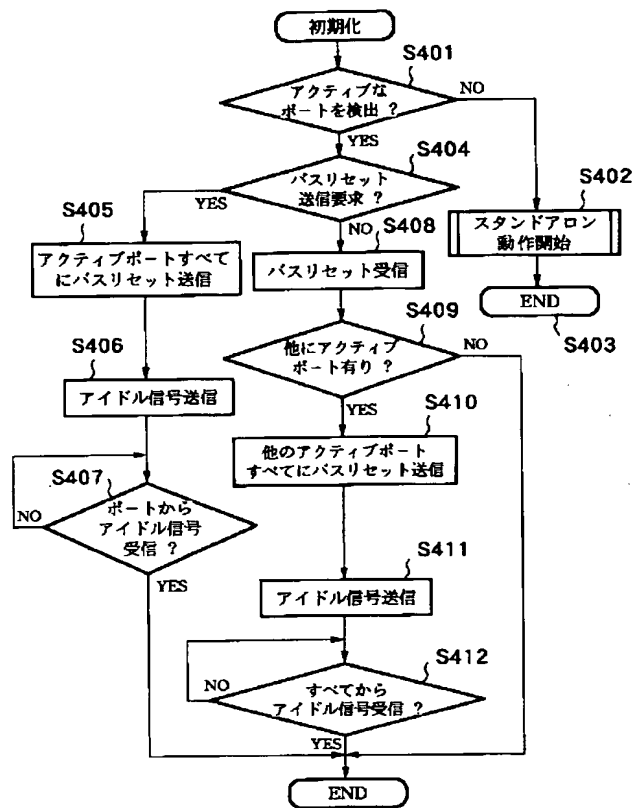
【図7 (c)】



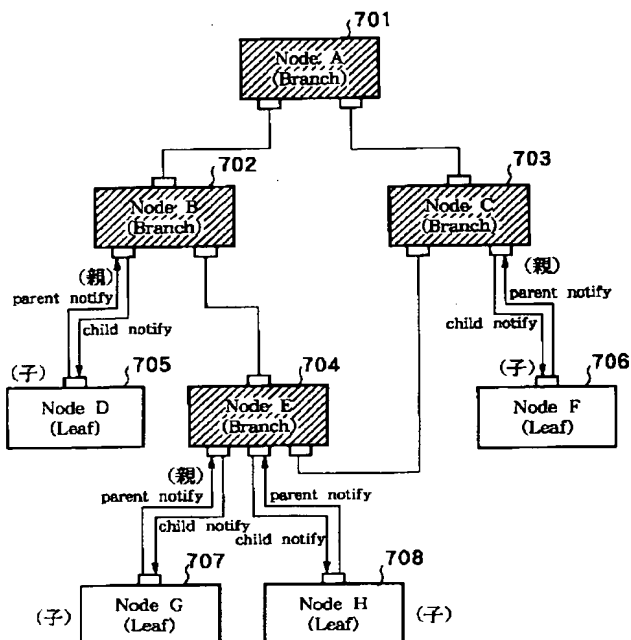
【図 3】



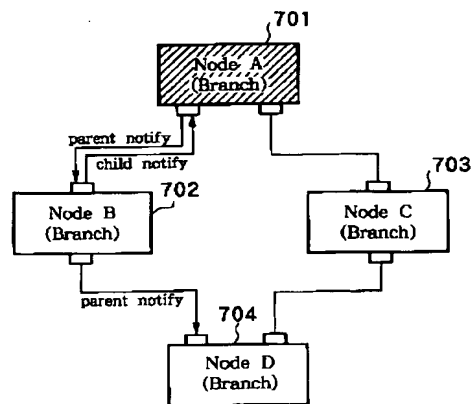
【図 4】



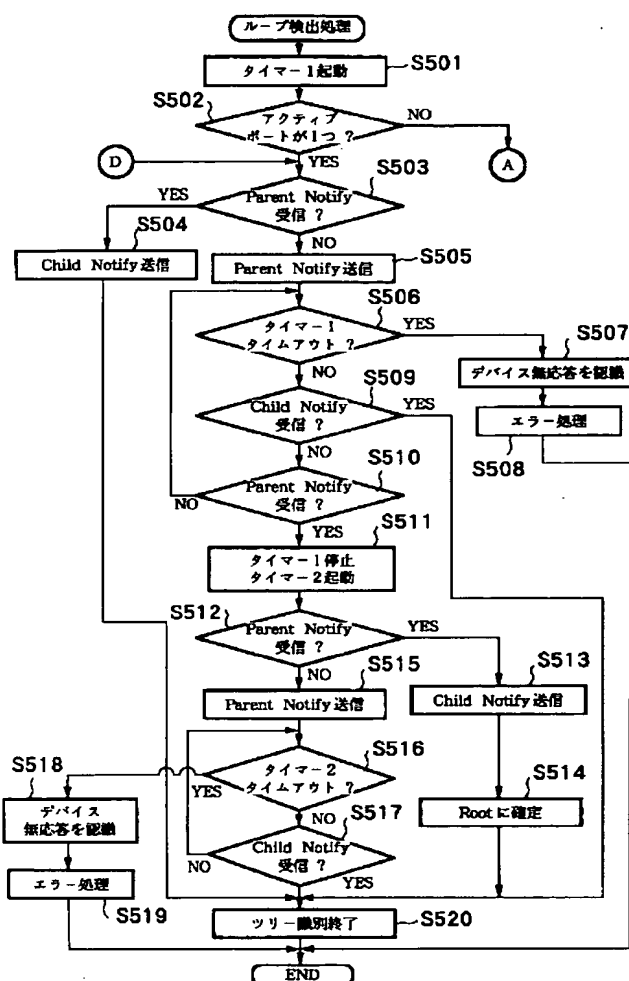
【図 7 (a)】



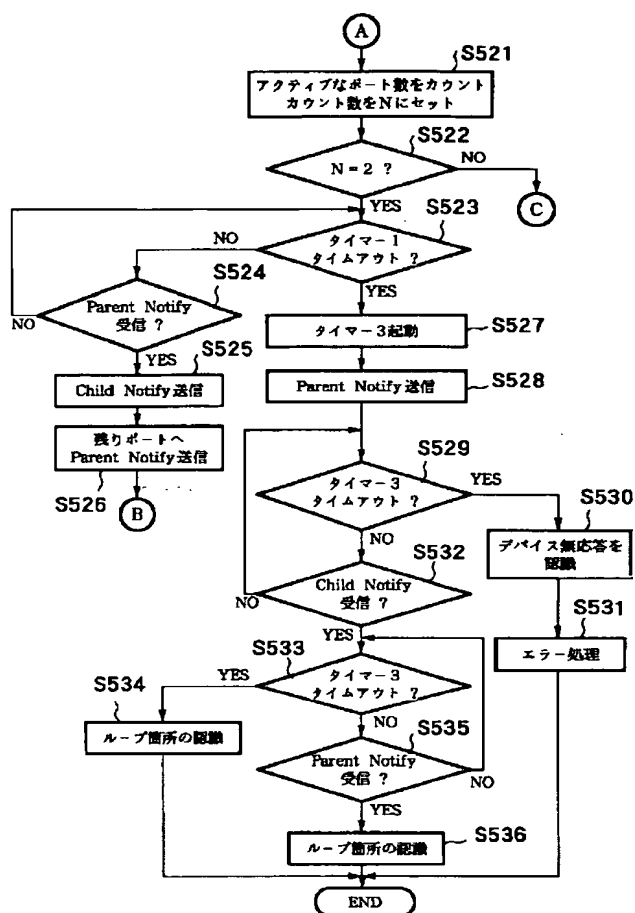
【図 7 (d)】



【図5 (a)】

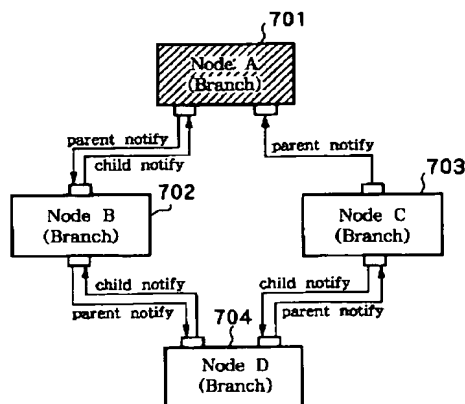


【図5 (b)】

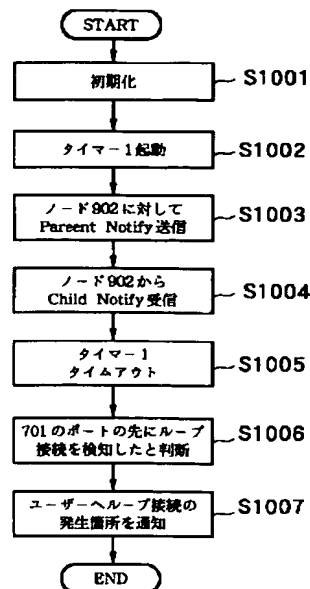
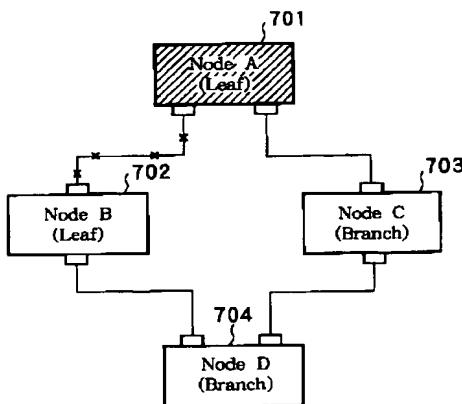


【図10】

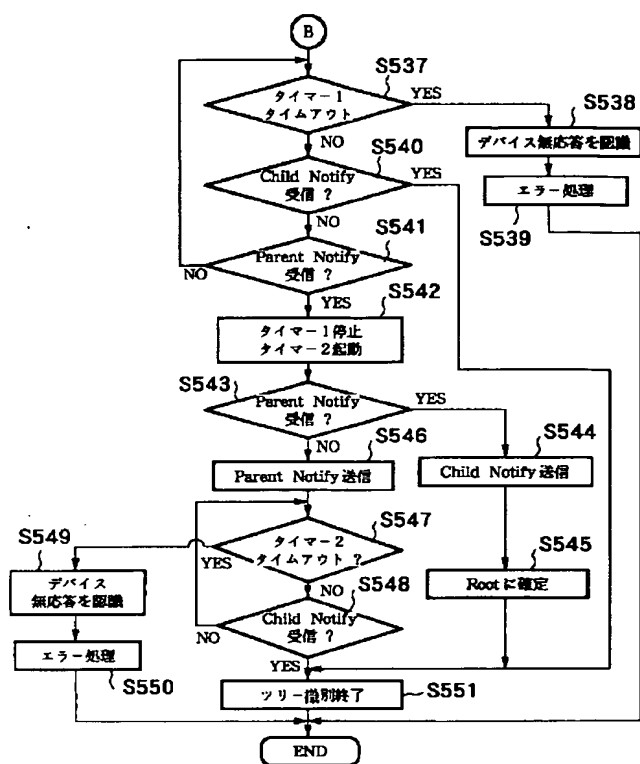
【図7 (e)】



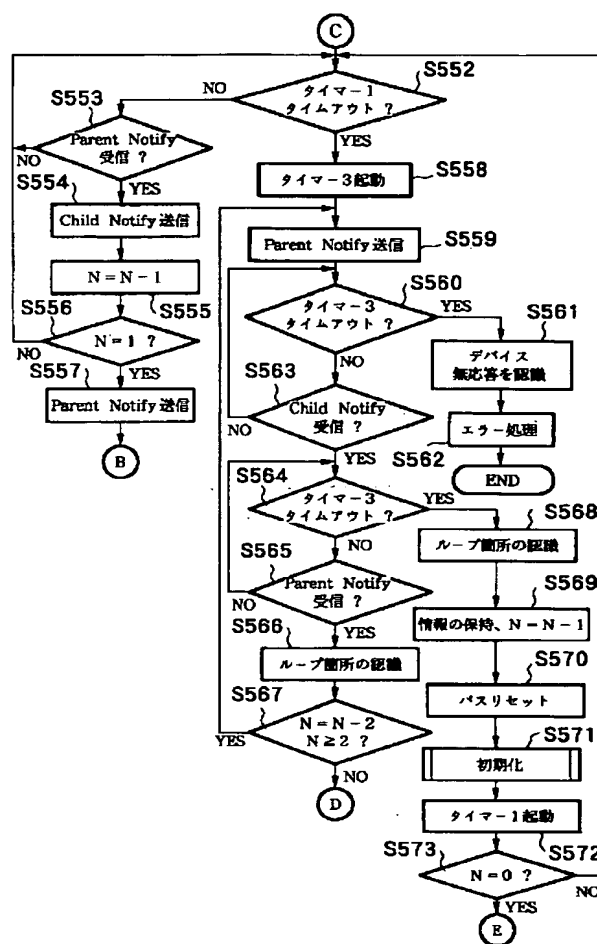
【図7 (f)】



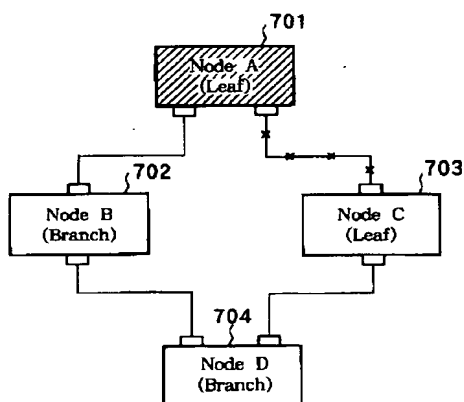
【図5(c)】



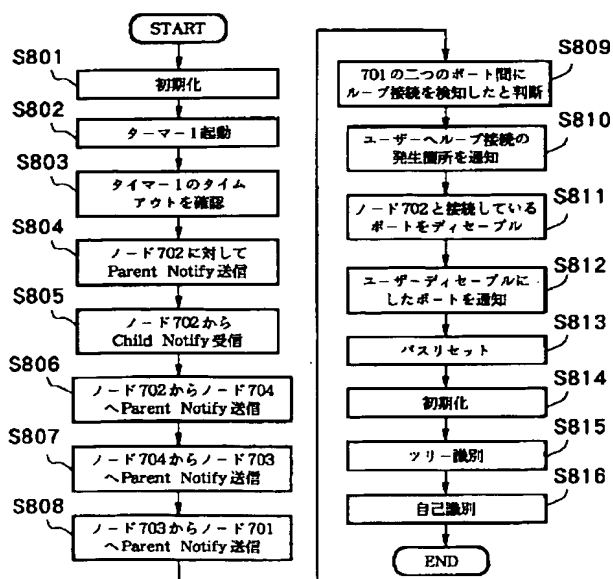
【図5(d)】



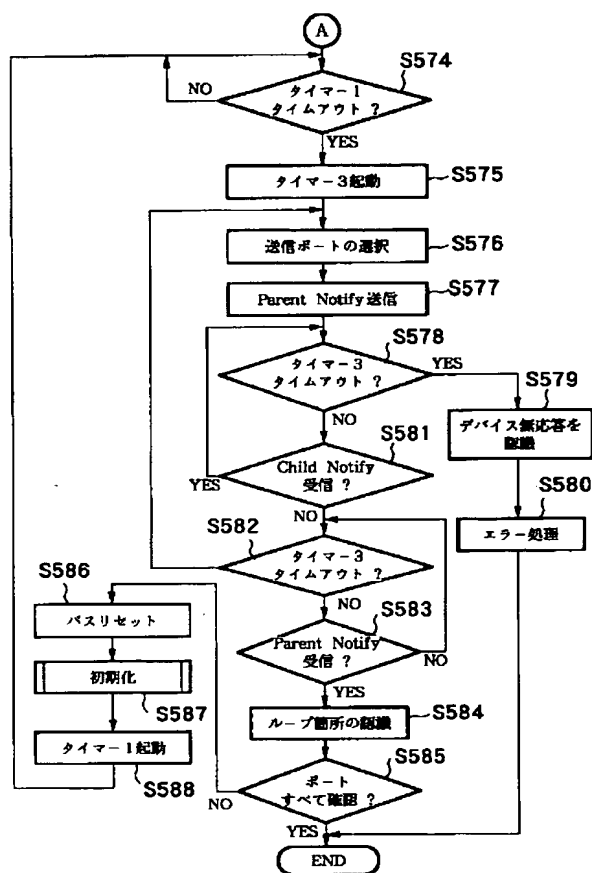
【図7(g)】



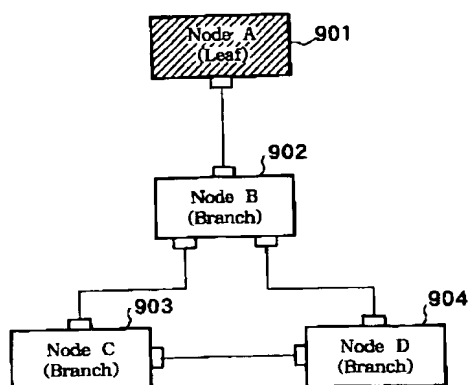
【図8】



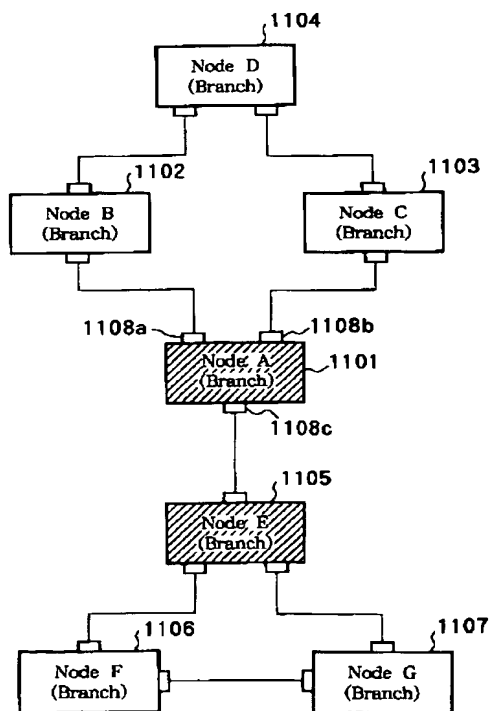
【図 5 (e)】



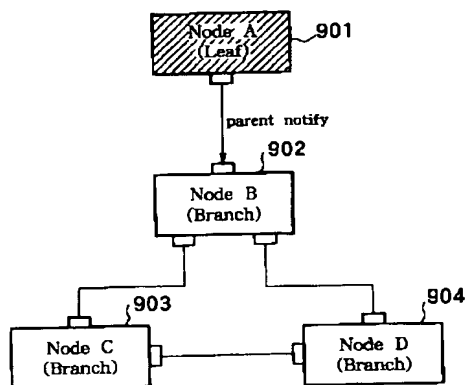
【図 9 (a)】



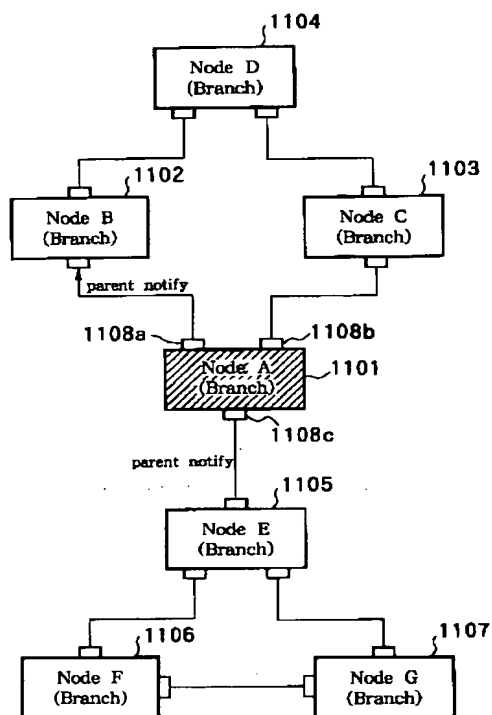
【図 1 1 (a)】



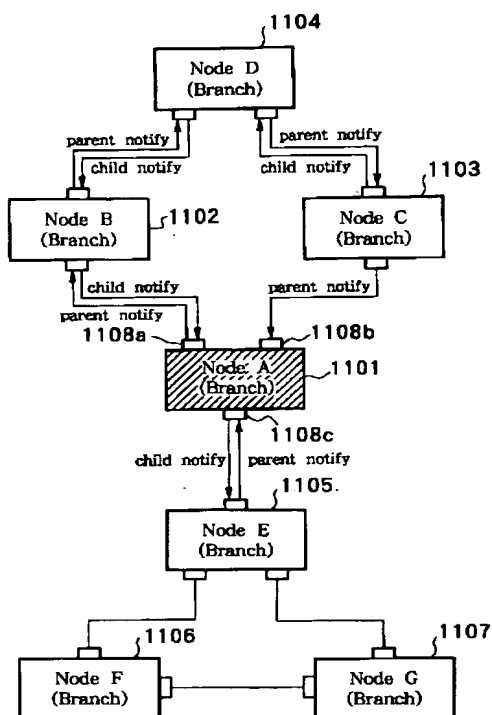
【図 9 (b)】



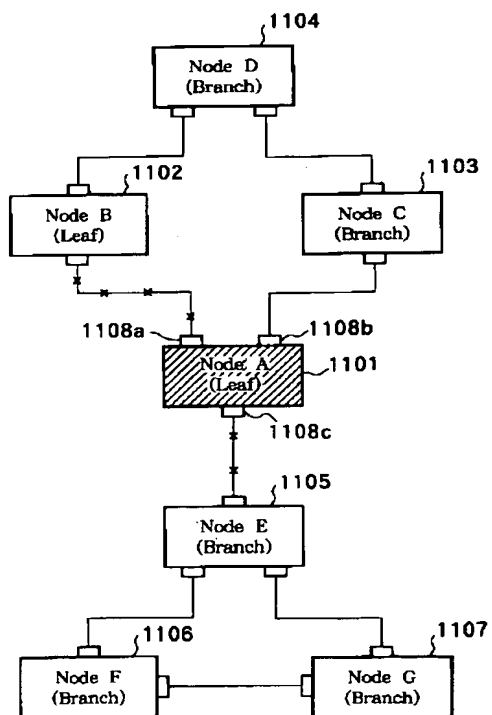
【図 1 1 (b)】



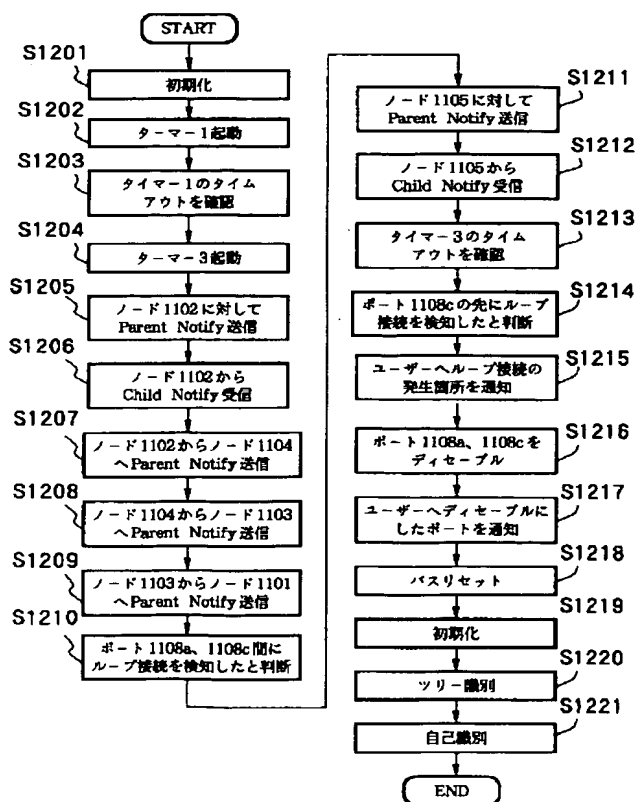
【図 1 1 (c)】



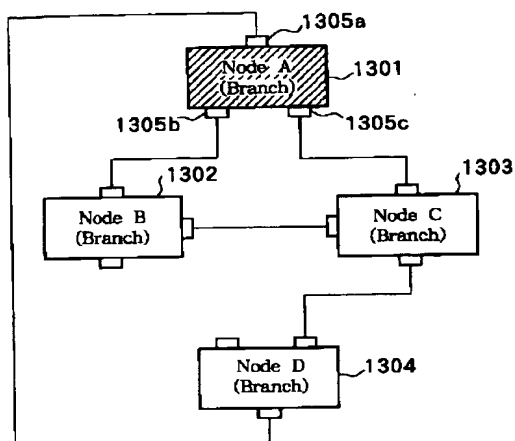
【図 1 1 (d)】



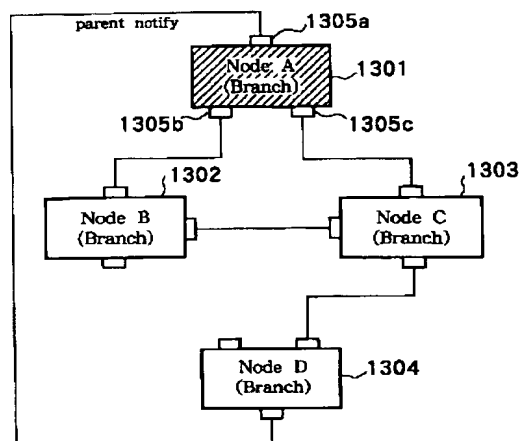
【図 1 2】



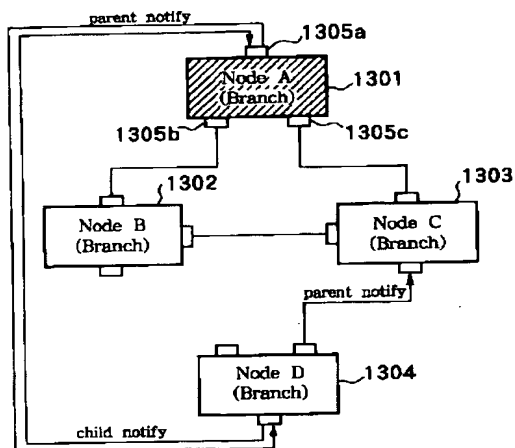
【図 13 (a)】



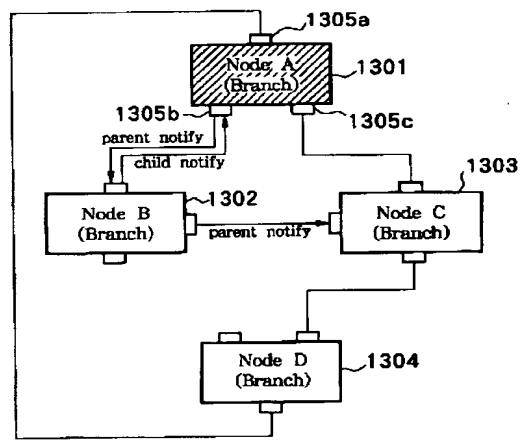
【図 13 (b)】



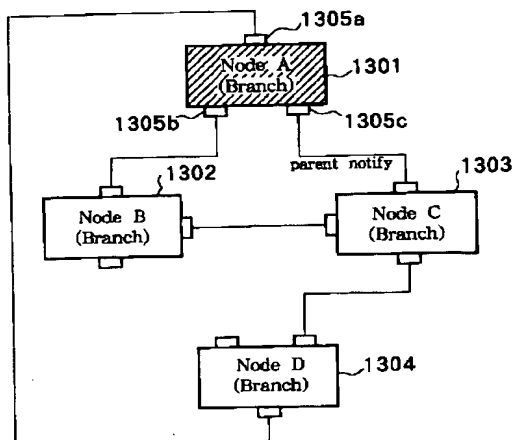
【図 13 (c)】



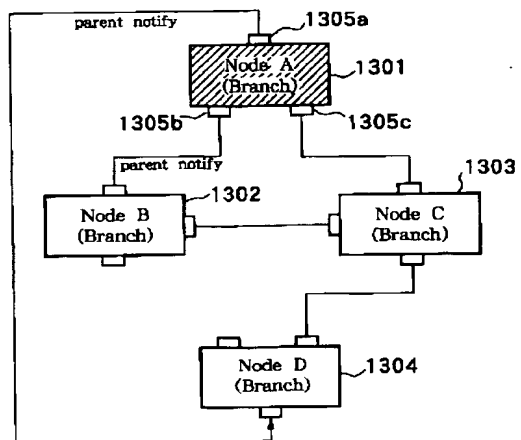
【図 13 (d)】



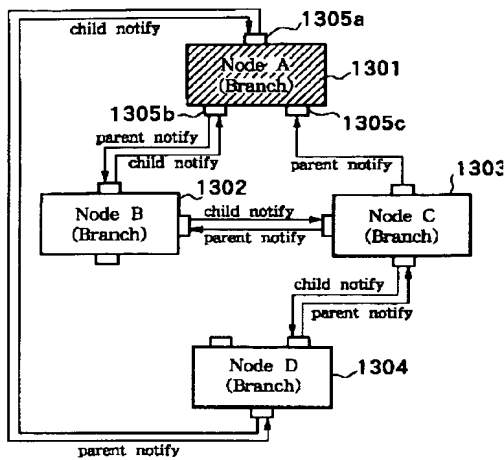
【図 13 (e)】



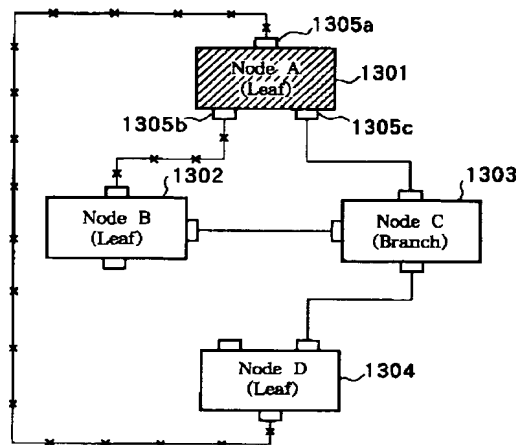
【図 13 (f)】



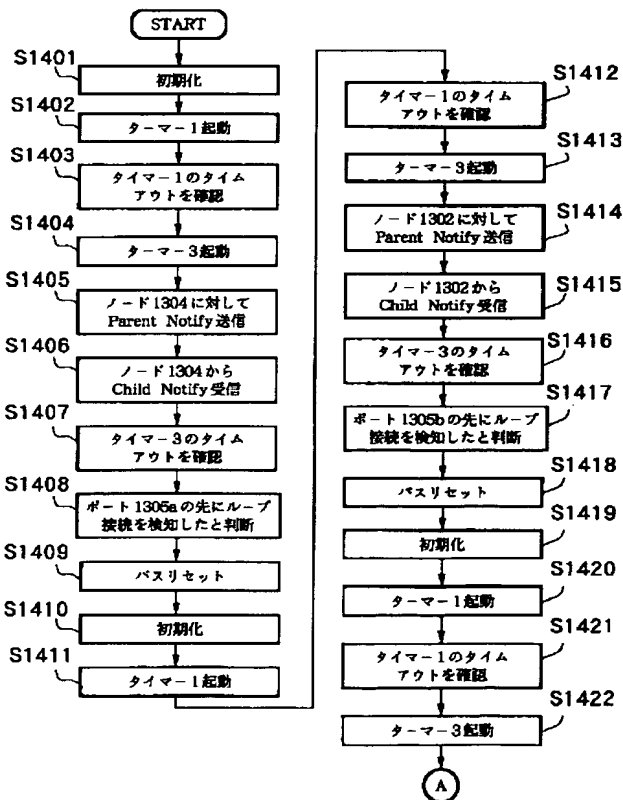
【図13 (g)】



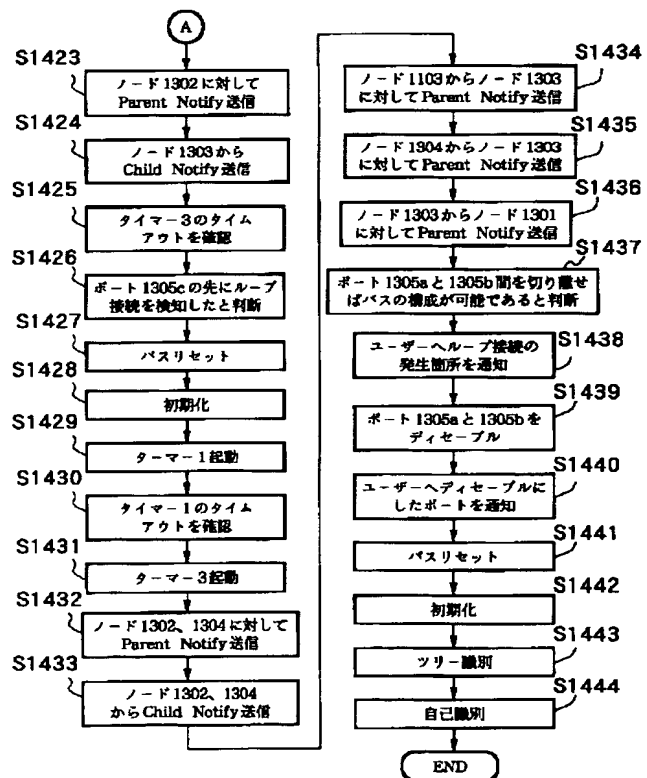
【図13 (h)】



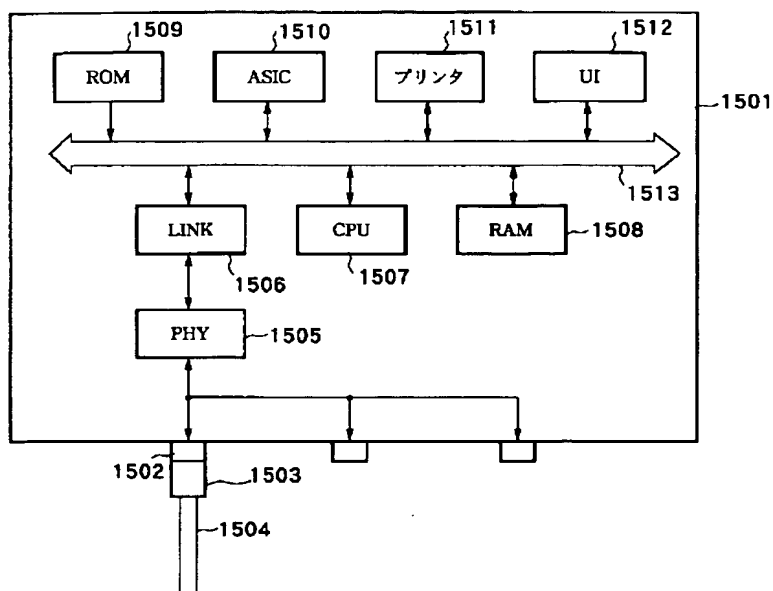
【図14 (a)】



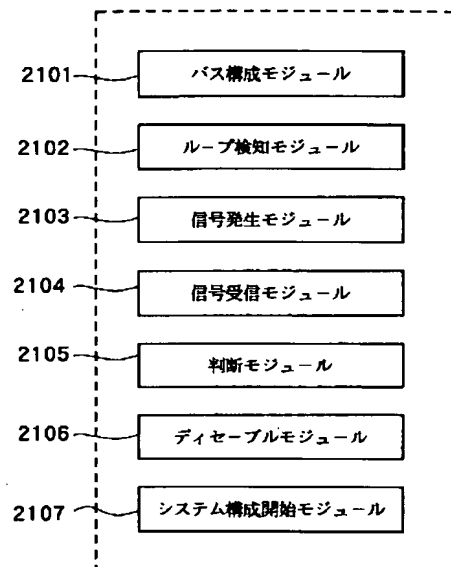
【図14 (b)】



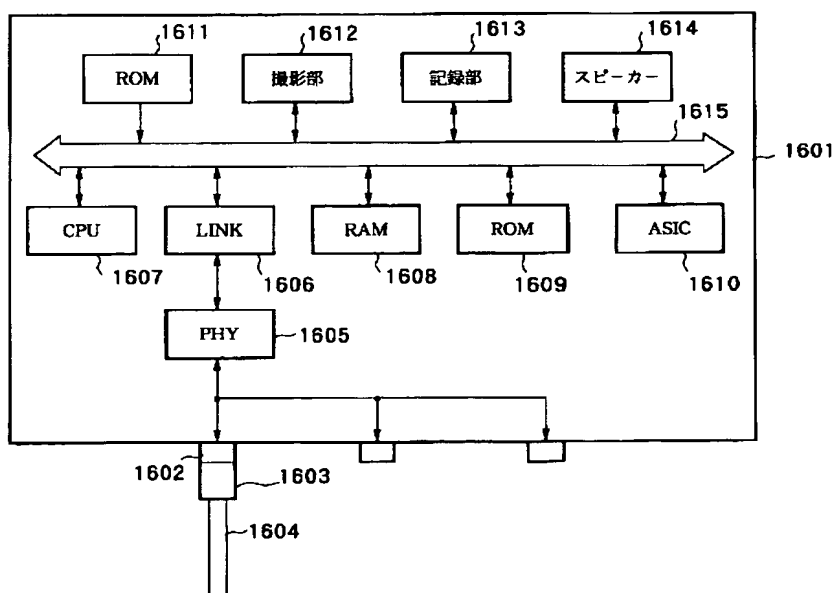
【図 1 5】



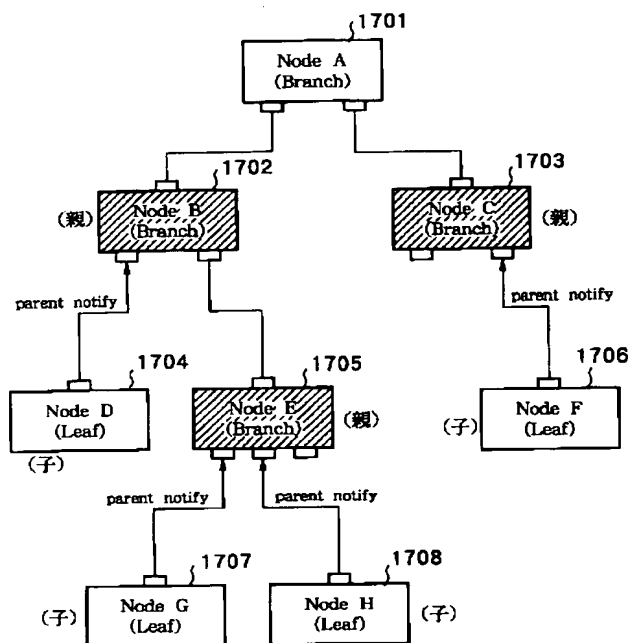
【図 2 1】



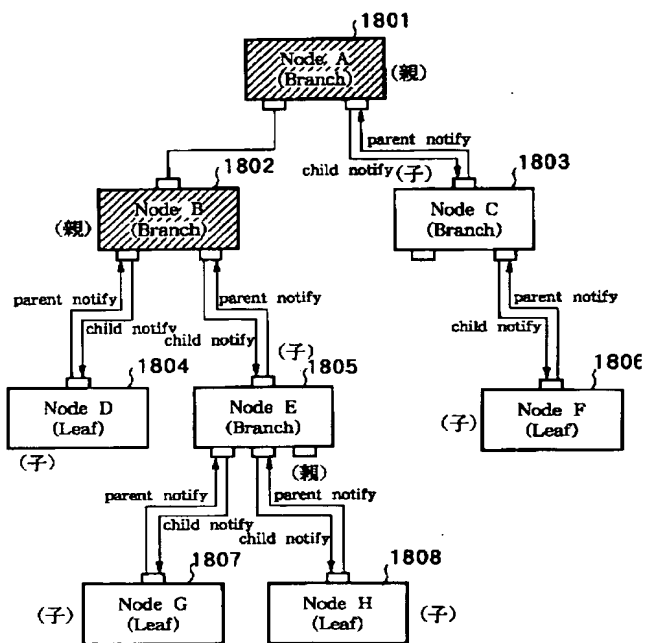
【図 1 6】



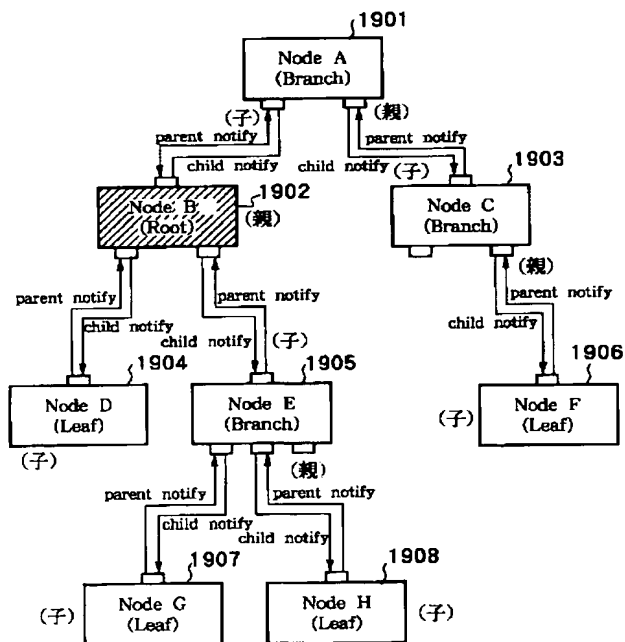
【図 17】



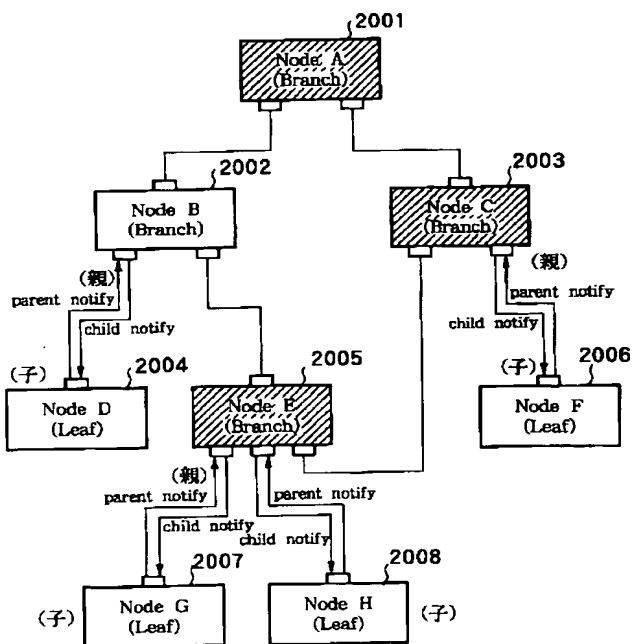
【図 18】



【図 19】



【図 20】



フロントページの続き

(51) Int. Cl. ⁶

H 0 4 L 12/28

識別記号

F I

H 0 4 L 11/00

3 1 0 D

This Page Blank (uspto)